12-2017

Predictive Maneuver Planning and Control of an Autonomous Vehicle in Multi-Vehicle Traffic with Observation Uncertainty

Qian Wang
Clemson University

PREDICTIVE MANEUVER PLANNING AND CONTROL OF AN AUTONOMOUS VEHICLE IN MULTI-VEHICLE TRAFFIC WITH OBSERVATION UNCERTAINTY

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Automotive Engineering

---

by
Qian Wang
December 2017

Accepted by:
Dr. Beshah Ayalew, Committee Chair
Dr. Andrej Ivanco
Dr. Yunyi Jia
Dr. Simona Onori

ABSTRACT

Autonomous vehicle technology is a promising development for improving the safety, efficiency and environmental impact of on-road transportation systems. However, the task of guiding an autonomous vehicle by rapidly and systematically accommodating the plethora of changing constraints, e.g. of avoiding multiple stationary and moving obstacles, obeying traffic rules, signals and so on as well as the uncertain state observation due to sensor imperfections, remains a major challenge.

This dissertation attempts to address this challenge via designing a robust and efficient predictive motion planning framework that can generate the appropriate vehicle maneuvers (selecting and tracking specific lanes, and related speed references) as well as the constituent motion trajectories while considering the differential vehicle kinematics of the controlled vehicle and other constraints of operating in public traffic. The main framework combines a finite state machine (FSM)-based maneuver decision module with a model predictive control (MPC)-based trajectory planner. Based on the prediction of the traffic environment, reference speeds are assigned to each lane in accordance with the detection of objects during measurement update. The lane selection decisions themselves are then incorporated within the MPC optimization.

The on-line maneuver/motion planning effort for autonomous vehicles in public traffic is a non-convex problem due to the multiple collision avoidance constraints with overlapping areas, lane boundaries, and nonlinear vehicle-road dynamics constraints. This dissertation proposes and derives some remedies for these challenges within the planning framework to improve the feasibility and optimality of the solution. Specifically, it

introduces vehicle grouping notions and derives conservative and smooth algebraic models to describe the overlapped space of several individual infeasible spaces and help prevent the optimization from falling into undesired local minima. Furthermore, in certain situations, a forced objective selection strategy is needed and adopted to help the optimization jump out of local minima.

Furthermore, the dissertation considers stochastic uncertainties prevalent in dynamic and complex traffic and incorporate them with in the predictive planning and control framework. To this end, Bayesian filters are implemented to estimate the uncertainties in object motions and then propagate them into the prediction horizon. Then, a pair-wise probabilistic collision condition is defined for objects with non-negligible geometrical shape/sizes and computationally efficient and conservative forms are derived to efficiently and analytically approximate the involved multi-variate integrals. The probabilistic collision evaluation is then applied within a vehicle grouping algorithms to cluster the object vehicles with closeness in positions and speeds and eventually within the stochastic predictive maneuver planner framework to tighten the chanced-constraints given a deterministic confidence margin. It is argued that these steps make the planning problem tractable for real-time implementation on autonomously controlled vehicles.

DEDICATION

To my wife, Shilun, my dad, Darao, my mum, Meng and the other family members for their love, support and encouragement in my journey towards the PhD degree.

ACKNOWLEDGMENTS

I would like to express my deep gratitude to Professor Beshah Ayalew, my advisor for his patient and consistent guidance, enthusiastic encouragement and valuable critiques on this dissertation and my study in CU-ICAR. I would also like to offer my special thanks to Dr. Thomas Weiskircher and Dr. Ardalan Vahidi for their important guidance and help on the initial work of this dissertation. My grateful thanks are also extended to my advisory committee members, Dr. Simona Onori, Dr. Yunyi Jia and Dr. Andrej Ivanco for their helpful suggestions.

I also wish to thank my colleagues, Chunjian Wang, Jeffery Anderson, Xiaoqing Cao, Adamu Yebi, Sadadru Dey, in our research group and Zhe Wang, Bin Xu, Zhiyuan Du, Darui Zhang, Zeren Xu, Xiaoyan Yu, Shichao Huo, Siyu Guo, Xueyu Zhang and Zifan Liu at CU-ICAR, Clemson. I appreciate their continuous encouragement and help to my research and life.

TABLE OF CONTENTS

Page

Table of Contents (Continued) Page

LIST OF TABLES

LIST OF FIGURES

List of Figures (Continued)

Figure Page

List of Figures (Continued)

Figure Page

List of Figures (Continued)

Figure Page

List of Figures (Continued)

Figure Page

List of Figures (Continued)

Figure Page

CHAPTER 1

INTRODUCTION

The autonomous vehicle is right on the road of development to change our future mobility. It has a huge potential for boosting the safety of driving and freeing the driver from driving efforts due to the advanced sensing equipment and control algorithm [1] [2] [3]. It's also likely to improve the efficiency of road transportation systems via safe increases in traffic density, minimize pollutions and energy waste through reducing congestion and help the disable people for daily transportation.

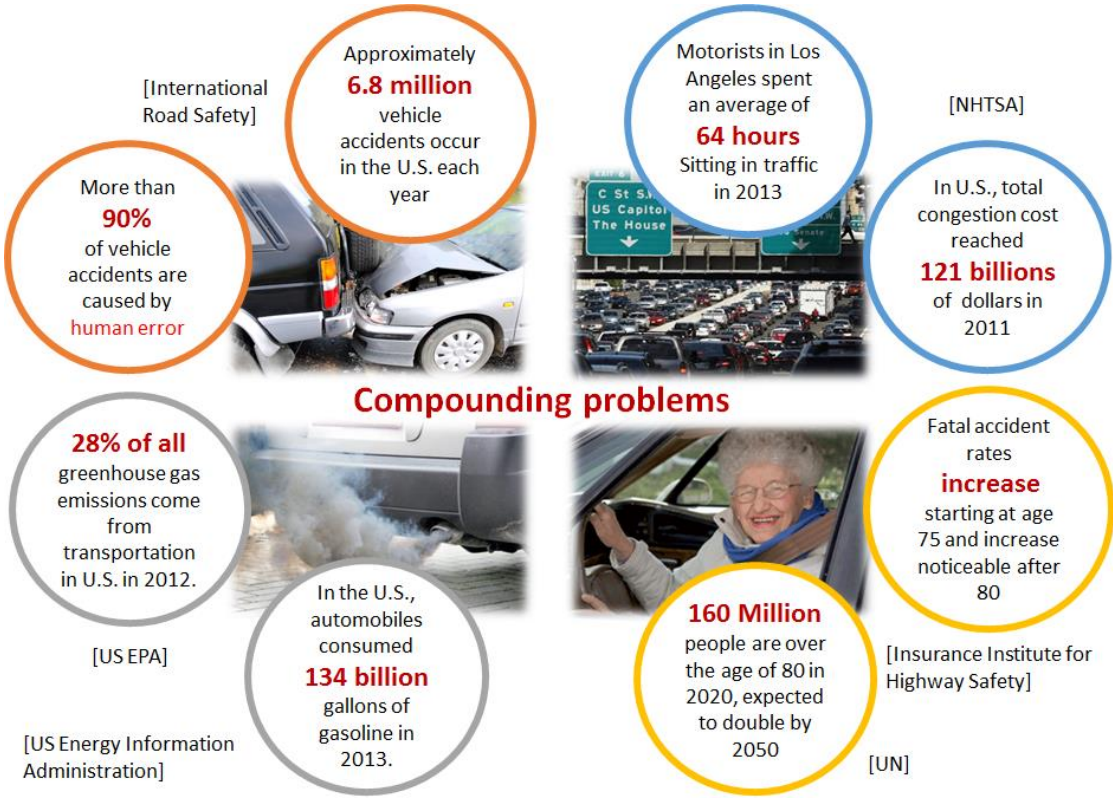


Figure 1-1: The compounding problems that can be potentially solved by autonomous vehicle

1.1 Overview of the Control Problem for Autonomous Vehicle

The basic problem for the control of autonomous vehicle is to determine the mapping from the sensing data to the control signal of the actuators that ensures a reliable, safe, legal and comfortable drive from the starting place to the final target place. The sensing data are obtained from the perception sensors like camera, li-dar, radar that responsible for environment detection and the monitor sensors like inertial motion sensor and GPS that observes the ego vehicle states. The actuators are manipulated by the drive by wire system that includes steering, braking and throttle control of the vehicle.

Recently, there exists two main approaches in different direction to solve the problem defined above: 1) Learning based control, 2) Planning based control.

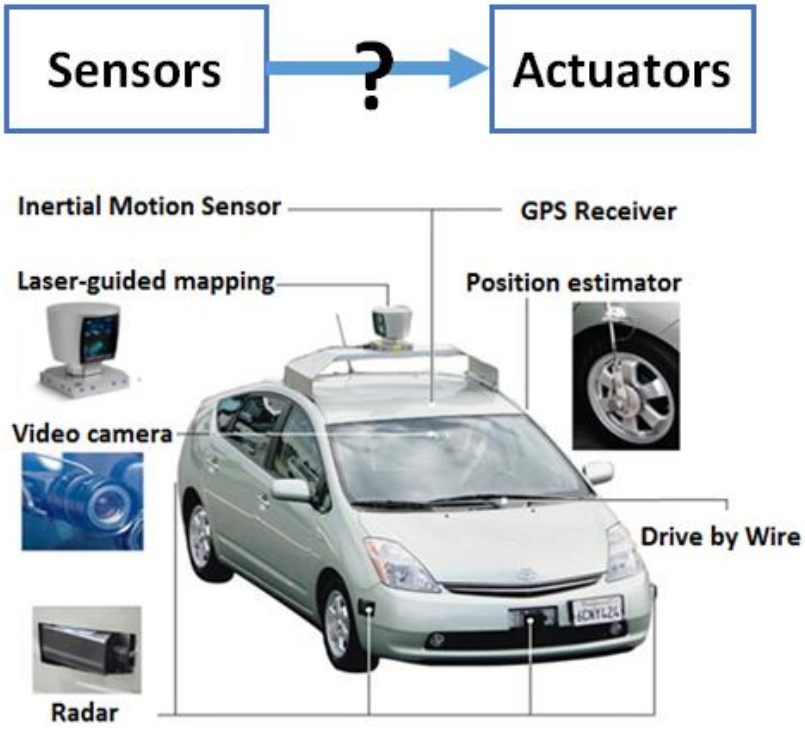


Figure 1-2: Control problem of the autonomous vehicle

Learning based control uses supervised learning method to train a neural network (NN), which is a policy function, based on the pair data records of input (sensing data/machine vision results) and output (driver's control signal) to the test vehicle, and then apply the NN on the ego vehicle for control purpose. This approach dates back from 1980s and it tries to imitate a human driver's reaction according to the sensing input, therefore it's also called imitation learning.

Planning based control primarily originates from the robotics field and follows a manually designed decision making pattern of the vehicle motion based on the navigation and situation understanding information from the navigation system and environment perception system to control the ego vehicle. The approach is widely applied in the development of the autonomous vehicle.

These two approaches will be described in detail in the following sections.

1.1.1 Learning Based Control

Based on the preprocessing level of the sensing data, the imitation learning can be divided in to two categories: 1) End to end learning and 2) Perception based learning.

In the end to end learning, a direct and simple mapping from the sensory input, such as a front face camera, to actuations necessary for driving, like the angle of steering input,

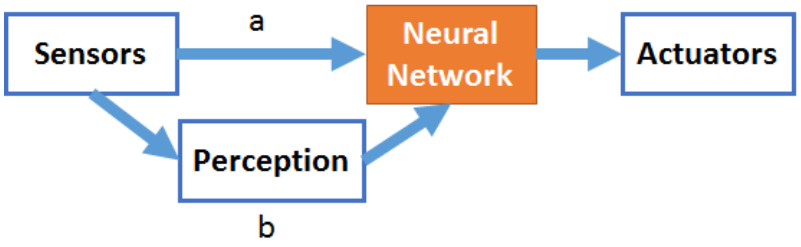


Figure 1-3: Illustration of end to end learning (a) and perception based learning (b)

braking and throttle. This idea dates back to the late 1980s, when [4] uses a NN that takes input from a front facing camera and a range detect sensor and returns a quantized steering wheel angle. The training data was collected form simulation. A similar approach was taken later by [5] in 2005 to train a convolutional NN (CNN) to drive an off-road mobile robot. Recently, [6] used similar but deeper CNN for steering control in lane following with three front camera input in training process and single camera in application. The performance is demonstrated via real road test. [7] proposed a query efficient way to improve the training performance for CNN based on the data collected from a racing video game. In this approach, full and direct information will be taken into training, which causes global performance and less process steps. But the redundant information unrelated to the driving will also increase the difficulty of CNN in recognition. Moreover, there is no clue of the inner decision making, it's hard to analyze the key factors that influence the actuation generation.

In the perception based learning, the sensing data will be processed by an learning based perception module to identify the key indicators of the local relation between the ego

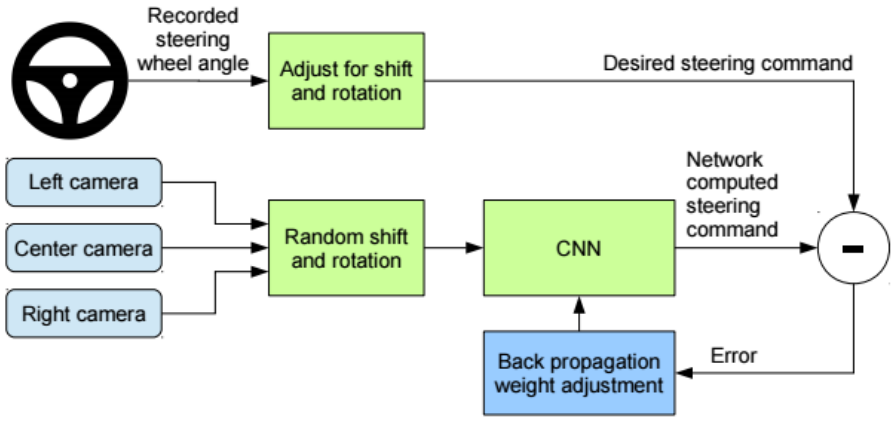


Figure 1-4: Training process of the CNN in end to end learning

vehicle and environment, which can be either a space description of the world model [8] [9] like lanes, traffic signals, object cars and pedestrians or the affordance but meaning full indicators [10] such as the heading angle error between the vehicle and road, the distance to the lane mark and the distance to the surrounding object cars. Then a simpler CNN can be used to map those indicators to the actuations. The perception module increases the processing steps but reduce the effort of CNN. The performance of this approach is highly limited by the perception accuracy.

There is no strong evidence that which approach is better as they all have their own pros and cons as mentioned. However, as they all trying to imitate a human driver, the perception based learning approach is more close to a human decision on which the influence of the driving related indicators are considered as input of the policy function in NN. Even though it seems like a permanent solution for the autonomous vehicle that an artificial human driver replace a real human driver, there is still a long way for this approach to overcome the challenge from tremendous data collection, diversity of driving style, reliability under complex scenarios.

1.1.2 Planning Based Control

Aside from the learning based control, planning based control follows a more reasonable and feasible pattern that relied on the decision making in a sequence of activities required to achieve a desired goal. This approach generally includes the following components between the sensors and actuators: 1) Perception, 2) Planner and 3) Executor.

Similar like the perception module introduced in the learning based approach, it's responsible for transforming the raw sensor data into a cognitive world model that helps

Figure 1-6: Illustration of planning based control

the planner to make a planning decision. The perception consist of the environment detection [11] [12] such as identifying the lane, traffic signals and the object tracking of the cars and pedestrians as well as the ego vehicle condition observation [13] [14]such as localization and vehicle state estimation. The challenges are reliable perception algorithm design and handling the uncertainties in sensor measurement.

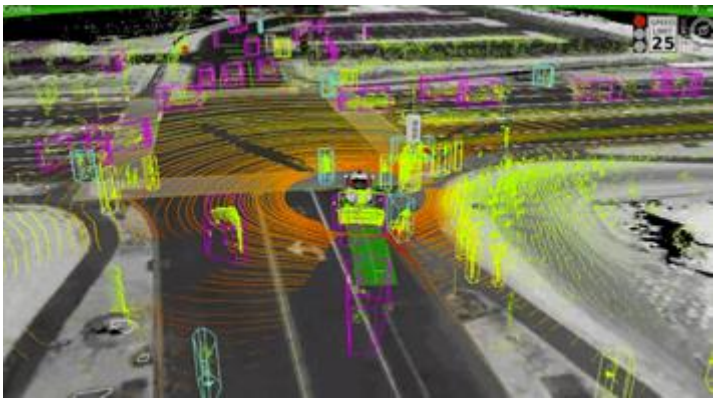


Figure 1-5: A typical perception results from the sensing data by Google

The planner is responsible for the decision making on the feasible activities planned to satisfy the desired goals. Depending on the level of the goal, the planner can be design in to a hierarchical fashion:

- Global/long term planning: route planning [15] [16]
- Local/short term planning: motion planning

Figure 1-7: Illustration of route planning (left) and motion planning (right)

In the higher level, the route planning works like navigation system that intends to find the best way points from position A to B in the global sense based on the digital map, traffic and localization information. It doesn't consider the detail information of the traffic but only the roads selected to connect A and B, In the lower level, the motion planning aims to plan the motion trajectory in a short future horizon of the vehicle in the local environment with limited information to abbey the constrains like following the traffic rules, yield to obstacle to avoid a collision, satisfying the vehicle dynamics and so on. The motion plan will be update fast enough to accommodate the dynamics change of the environment. This proposal focus on improving the solution of motion planning problem, details will be introduced in the next chapter.

The executor ties generating the desired control signal that can guide the vehicle to follow the decision of the planner. It's usually a tracking controller that considering the vehicle dynamics like the normal vehicle dynamics controller. In some cases, the executor will be integrated inside the planner that the control signal also become part of the plan.

Overall, the planning based control makes decision on the motion and control plans based on the pre-programmed algorithm according to information from the pre-stored map data and perception. Its performance is limited to the quality of the perception and the planning algorithm as well as the dynamic uncertainties of the environment and the ego vehicle.

1.2 Motion Planning for Autonomous Vehicle

1.2.1 Problem Definition

As mentioned, the motion planning aims to plan the motion trajectory in a short future horizon of the vehicle in the local environment with limited information to satisfy several constrains, like following the traffic rules, yield to obstacle to avoid a collision, satisfying the vehicle dynamics and so on. The literature about this problems will be reviewed and summarized to find the gap/motivation of this dissertation work.

1.2.2 State of the Art

The majority of the existing researches in motion planning are found in the robotics field, where various algorithms are proposed to find collision-free trajectories under static and dynamic constraints in the available space. In this discussion, the word motion is used to mean state trajectories (including both discrete state like the maneuvers acceleration, deceleration, steering and the continuous state, at a minimum, position or path and speed) for the controlled vehicle. The state of the art in planning methods roughly falls into three groups: 1) Sampling based method, 2) Decomposed method 3) Mathematical programming based method.

In the sampling-based methods, the state and/or input space is discretized or randomly sampled in lattices and then efficient heuristics for deterministic or stochastic searching, such as the A* graph search or RRT* algorithm are applied to find the best collision free trajectory based on an objective function [17]. This approach can be used to design the maneuver-based motion planning/maneuver automaton, which is a currently popular method for guidance of robotic vehicles as it accommodates the practical vehicle maneuvers and simplifies the planning work with low on-line computational time by searching from a finite number of quantized motion primitives in a pre-defined library [18]. The motion primitives contain a library of steady-state and transient state trajectories [19] [20] connecting two steady-states. These primitives are generated by numerically solving the nonlinear vehicle dynamics model. Either random or deterministic searching methods, such as rapidly exploring random tree* based algorithm [21], particle swarm optimization algorithm [22]or greedy search algorithm [23], can be applied in real-time to construct a periodic planning law from the library, ensuring some robustness in a disturbance environment. However, the flexibility and effectiveness of the control strategy is a function

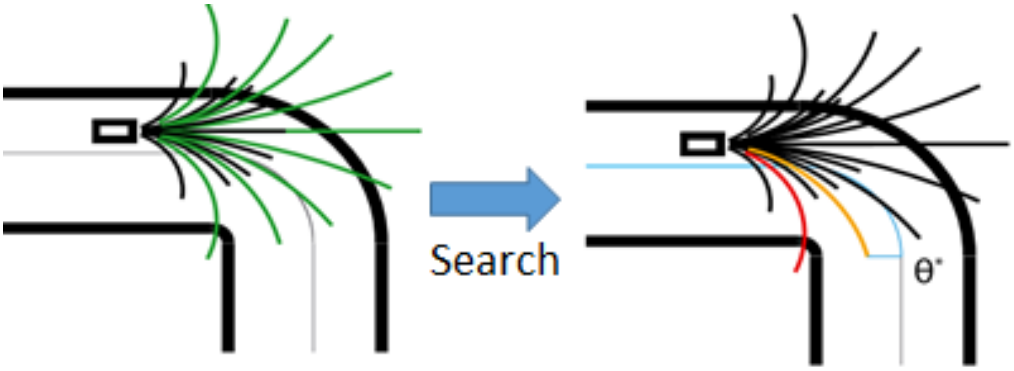


Figure 1-8: Illustration of motion primitive selection under different velocities and steering [20]

of the resolution and size of the library. Moreover, the complexity of finding the best maneuver increases with the resolution of the library.

In decoupling methods, the planning problem is usually decomposed into two easier sub-problems: first, applying a path planner (could be based on cell decomposition as in [24], or a sampling-based method) to find the waypoints in the configuration space, considering the shape of the ACV, and then using a close-loop controller to track those waypoints. The differential constraints are typically only applied to the latter sub-problem. Nevertheless, it's hard to prove the existence and the optimality of the collision-free solution, especially in the presence of uncertainties.

A third group of planning algorithms involve mathematical constrained optimization formulations which offer some guarantees of conditional existence and optimality of the solution based on the convexity of the problem formulation and the quality of initial guesses. In this group, receding horizon control (RHC) or model predictive control (MPC)-based motion planning generates optimal trajectories by repeatedly solving

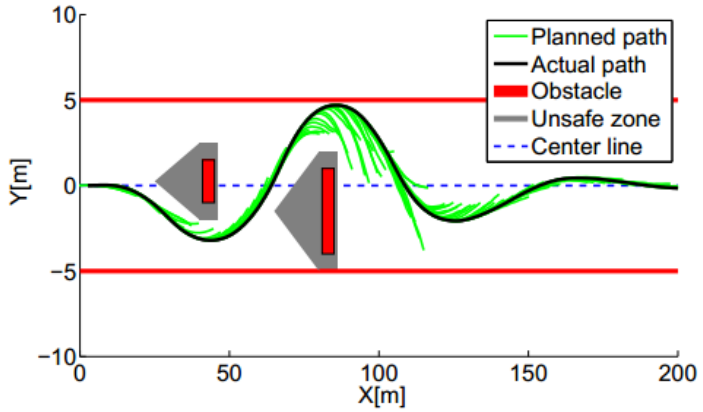


Figure 1-9: Illustration of MPC based motion planning [29].

constrained optimization problems in a receding prediction horizon [25]. It is very attractive because it naturally handles multiple constraints and modeled nonlinearity in the full space region. When applying MPC to vehicle guidance problems a tradeoff is usually made between the model fidelity and the computational complexity. In [26], the computational burden of using complex vehicle nonlinear dynamics as well as the tightened constraints such as needed for obstacle avoidance was found troublesome. Model simplification was then adopted to reduce the complexity of solving the planning problem. A linear time-variant (LTV) vehicle model was used in [27] to approximate the vehicle dynamics, which showed feasibility for real time application at the cost of reduced prediction accuracy. In [28] [29] a two-stage hierarchical nonlinear MPC (NMPC) framework was proposed, where a low-fidelity nonlinear point-mass vehicle model is used in the top level MPC for motion planning with the global constraints like obstacle avoidance, friction ellipse, while the lower level MPC uses a high-fidelity vehicle dynamics model with local constraints like tire dynamics or actuator limits. This allows the nonlinear dynamics of the basic vehicle motion to be considered, but the scheme relies on model parameters that are likely to change (e.g. vehicle mass, inertia, tire properties). To improve the adaptability of the planning framework in public traffic, in [30] [31] a nonlinear particle motion model expressed in reference path frame is used to design a high-level MPC for motion planning. The reference path is defined by its curvature in polynomial form. Either the predicted control output or the predicted state trajectories were configured to be tracked using traditional vehicle dynamics controllers at the lower level. This approach showed good performance in dynamic path tracking and obstacle avoidance,

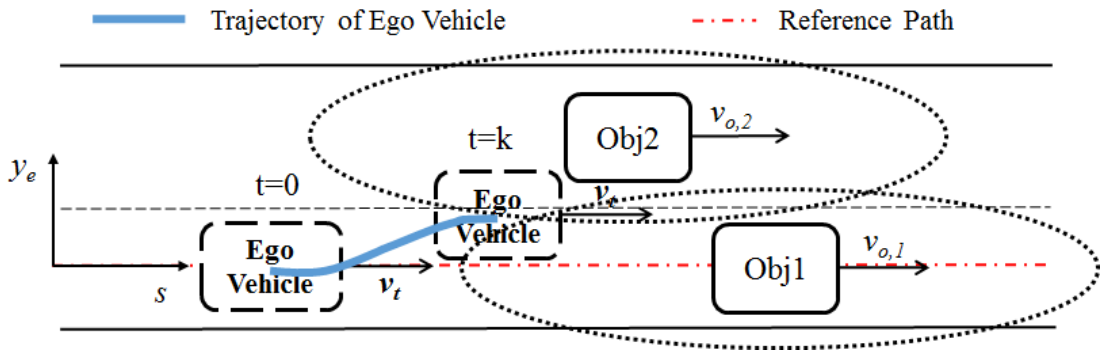


Figure 1-10: Illustration of a trapped ego vehicle with MPC based lane tracking and obstacle avoidance system

and the optimization problem can be overcome by efficient real-time optimization method [32] [33].

The MPC method mentioned above are independently implemented for specific maneuvers (e.g. ACC, path tracking or collision avoidance). However, a practical ego vehicle needs to have a maneuver planning framework to handle different situations. For instance, the ego vehicle should have the ability to decide whether to make an active lane change to pass a vehicle in front of it if there is available adjacent lane or merely follow the vehicle in front if not, otherwise some undesired behavior will happen. But these discrete decisions are too complex to implement in a single nonlinear MPC setup as the required computations have then to deal with hybrid system optimizations, which generally result in mixed integer programming problems and can require significant computation time [34] [35]. Some approach needs to be develop to solve this problem.

1.3 Research Motivation and Contributions

According to the literature review and our previous work on the MPC-based motion guidance system, the motion planning approach used in autonomous road vehicle is more

comprehensive in a hybrid (combined discrete maneuver planning and continuous trajectories planning and the interaction between them) manner. Unfortunately, the optimization problem of such a system (combining maneuver planning with motion planning) results in a mixed integer programming problem that is not feasible for real time application. Also, the non-convex configuration space make it hard for gradient based optimization method to find a better local minimum. Furthermore, when multiple object vehicles as well as the state uncertainties are involved, the original problem become intractable to solve. The dissertation is motivated by these problems.

The main contributions in this dissertation include:

- Propose a hybrid control framework that modelling the maneuvers of ego vehicle (tracking a specific reference speed on a specific lane) with particle description under predictive control as a hybrid control system
- Predictive reference speed assignment is applied in the system to improve the optimality of maneuver selection and based on the prediction of the object and ego vehicle states, thus realizing the maneuver planning.
- A relaxation method is applied to transform the mixed-integer programming problem into a nonlinear programming problem, which realizes the optimization based maneuver (for lane change) selection in real-time application.
- Propose an obstacle filtering algorithm to reduce the motion planning effort of the MPC and improve its optimality by generating a more conservative collision field for the group of object vehicles with close proximity, which helps to exclude the undesired local minima.

- Applied an evolution model to track the structure, state and boundary of the object vehicle group by consider the state distribution of each detected object vehicle.
- Derived the probabilistic collision/closeness criteria between any two individual object vehicles with non-negligible geometric size and shape information based on their state estimation via Bayesian tracking.
- Based on the closeness evaluation, a density-based method is applied to group/cluster the IOVs without a prior guess about the number of groups.
- Developed a constraint tightening strategy by deriving an analytical solution for the fundamental dimensions of the collision area according to the specified confidence threshold and the uncertainty distribution.
- Implemented and illustrated the constraint tightening method to a multi-vehicle grouping frame work and solved the stochastic MPC based motion planning problem for collision avoidance and traffic interaction.

1.4 Dissertation Organization

The dissertation is organized as follows. In Chapter 2, a hybrid predictive trajectory guidance and control framework is introduced to show the availability of the autonomous vehicle in switching among different maneuvers to accommodate the dynamic environment. In Chapter 3, the predictive maneuver planning and control framework is described to improve the optimality of selecting the maneuver in a predictive and optimized way. The relaxation method to convert the mixed integer programming to a nonlinear programming is also described in this chapter. In Chapter 4, obstacle filtering and vehicle grouping algorisms is design to facilitate the predictive maneuver planning via excluding the

undesired local minimums. In Chapter 5, a probabilistic framework for tracking the formation and evolution of multi-vehicle groups is introduced. The probabilistic collision/closeness criteria between any two individual object vehicles with non-negligible geometric size is derived. In Chapter 6, a constraint tightening approach is proposed to transfer the probabilistic collision avoidance constraint with a specified confidence threshold into a relevant deterministic constraint. The approach is then implemented to the multiple vehicle grouping framework and to solve the predictive maneuver planning problem with probabilistic collision avoidance constraint. Conclusion and future work are included in the final chapter.

CHAPTER 2

HYBRID PREDICTIVE TRAJECTORY GUIDANCE AND CONTROL FRAMEWORK

2.1 Abstract

In this chapter, a hybrid predictive trajectory guidance and control framework is proposed that enables the safe operation of autonomous vehicles considering the constraints of operating in dynamic public traffic. The core modules of the framework includes a nonlinear model predictive trajectory guidance (PTG) module that uses a computationally expedient curvilinear frame for the description the road and of the motion of the vehicle and other objects and an assigner module above it. The assigner module not only enforces constraints generated from information about obstacles/other vehicles/objects, public traffic rules for speed limits and lane boundaries, and the limits of the vehicle's dynamics, but also switches the references, e.g. reference lane and reference speed for the PTG module to follow based on the scenario or situation the ego vehicle is involved in. Therefore, the entire framework becomes a hybrid system. The performance of most aspects of the proposed scheme are illustrated by considering various simulations of the control framework application in typical public driving events, such as intersections, passing, emergency braking and collision avoidance. The feasibility of the proposed control framework for real-time application is highlighted with discussions of the computational execution times observed for these various scenarios.

2.2 Introduction

The basic problem in autonomous vehicle guidance is the planning and control of the motion trajectory to achieve the goal of moving from one location to another while fulfilling a number of constraints, which include: staying on the roadway, avoiding collisions with static or dynamic obstacles, obeying traffic rules, minimizing occupant discomfort from undesirable maneuvers [18]. An autonomously controlled vehicle (ACV) needs to rapidly and systematically accommodate these constraints and other environmental uncertainties.

In recent times, Model Predictive Control (MPC) has received significant attention for ACV motion planning due to its ability to readily handle input and state constrained optimizations on a prediction horizon that can then be implemented in a receding horizon scheme. Perhaps the simplest implementations of MPC designs in this area are those presented in [36], which primarily focused on the longitudinal dynamics and stability with adaptive cruise control (ACC) without considering the lateral vehicle dynamics. The works in [27] dealt with predictive trajectory/path tracking via single axle active steering inputs using nonlinear and linearized vehicle models, respectively. Only constant speed scenarios were considered; the longitudinal dynamics were ignored. A hierarchical two-level MPC framework was proposed in [28] to do predictive path tracking. A low fidelity model is used in the upper level MPC, and high-fidelity vehicle dynamics model was used in the lower level MPC. More stable results were observed in this case compared to applying only the lower-level MPC because of the already feasible trajectory reference generation by the

upper level MPC. This framework was later applied to achieve collision avoidance (CA) in [29], combining longitudinal and lateral vehicle dynamics control.

The MPC works mentioned above are independently designed for specific maneuvers (e.g. ACC, path tracking or CA). However, a practical ACV needs to have a multi-functional control framework to handle different situations. For instance, the ACV should decide to make an active lane change to pass a vehicle in front of it or merely follow the vehicle in front to obey public traffic rules. These discrete decisions are too complex to implement in a single nonlinear MPC setup as the required computations have then to deal with hybrid system optimizations, which generally result in mixed integer programming problems and can require significant computation time [37]. This could make them unsuitable for scenarios with fast dynamics.

One approach to address this is offered by considering a hierarchical framework where the discrete decisions of selecting maneuvers are relegated to an assigner module and a versatile MPC formulation handles the trajectory guidance in all or most possible maneuvers. The MPC in this PTG module integrates information about obstacles/other vehicles/objects and of public traffic rules for speed limits and lane boundaries, as well as limits of the vehicle's dynamics, in its constrained optimization. Therein, the assigner module is merely assumed to be available as an information filter that processes and delivers the data from the environmental information and vehicle dynamics sensors to the PTG.

In this chapter, we detail the functionality of the assigner module as one that manages the control setup of the MPC in the PTG. Here, the assigner is made a decision-

making module that guides the PTG to a specific maneuver state. A hybrid control framework is proposed with a (set of) finite state machine(s) (FSMs) designed for the assigner module, where the maneuver states are taken as the discrete states. The FSMs help the assigner to complete the task of not only processing external information but also choosing the desired maneuver state of the ACV. Then, the MPC of the PTG will be responsible to follow the chosen maneuver and generate the control input for the actuators available on the vehicle (steering, brake, traction).

A hybrid controller design for autonomous vehicles could be found in plenty of previous works. A hierarchical FSM concept with meta-state machine for different scenarios and a sub-state machine for vehicle maneuver states was designed in [38]. This FSM structure was further detailed by [39] with rule-based or Hidden Markov Method-based switching conditions to estimate human driver decisions. In [40], a rule-based automaton (FSM) was designed to regulate the longitudinal motion of ACVs to avoid collision under cruising and merging scenarios. A game theory method was used in [41] to design a robust hybrid controller, which guarantees safety under some uncertainties in vehicle platooning. The method was applied later by [42] with non-deterministic automaton to regulate an intersection problem. However, those works above were not interfaced with MPC-based trajectory guidance as we propose here.

The advantages of such a hybrid system view of the assigner are two-fold: First, with an exhaustive list of the maneuver states, one could cover all the basic functional behavior of the ACV to robustly react to environmental uncertainties [38], [41]. Second, as an agent of the transportation system, ACV can be made to react properly and

predictably with other vehicles via a proper and unified maneuver switching condition designed to preserve traffic order and efficiency [40], [41], [42].

2.3 Framework Design

Figure 2-1 shows a schematic of the proposed hierarchical hybrid control framework for an autonomous road vehicle. It consists of four modules: the route navigator module, the environment recognition module, the higher-level maneuver selection module and the lower-level maneuver execution modules. The details of Navigation and

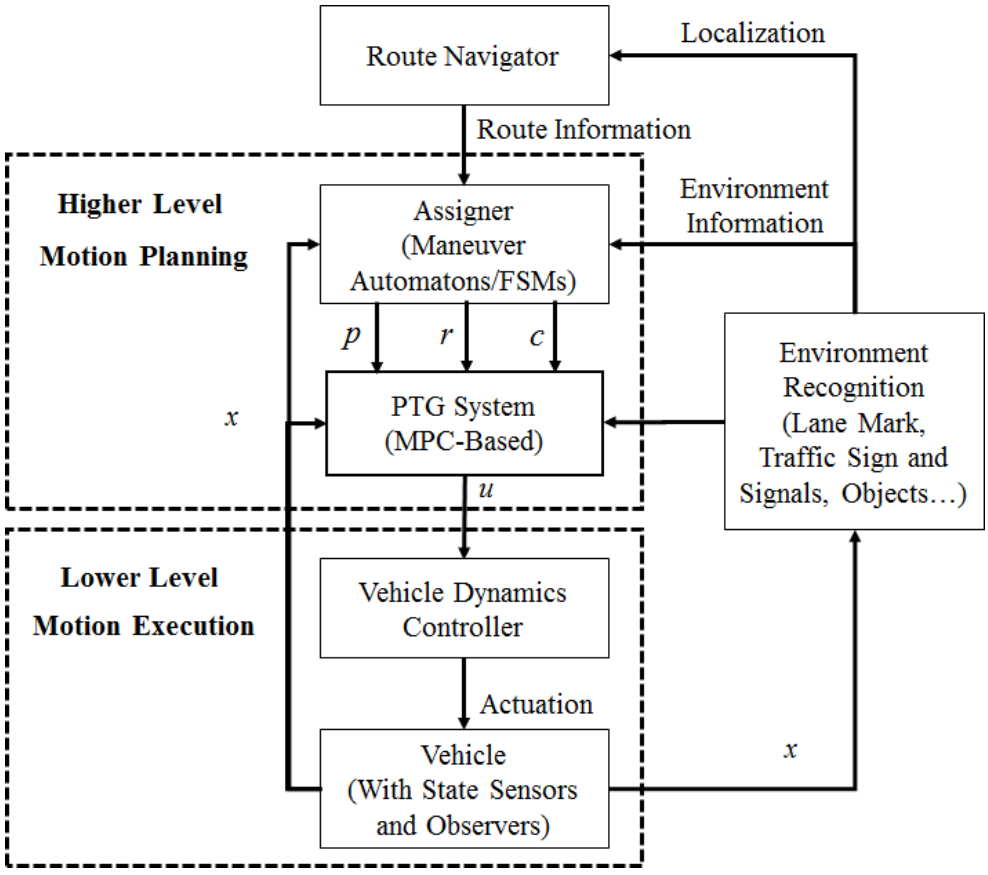


Figure 2-1: Hierachical control framework

environment perception module are beyond the scope of the work and their output information is simply assumed available to the maneuver planner.

The assigner module is responsible for processing the information from the route navigator and environment recognition and then provide the necessary information, e.g., parameters (p), references (r), and constraints (c), for the MPC formulation. Further description of these pieces of information can be found in [31]. More importantly, the maneuver automatons/FSMs are also designed and stored in this module. Based on the scenario the ACV is in, a relevant FSM will be chosen. The maneuvers are defined as tracking a suit of references such as a reference speed on a reference lane via recursive loop of MPC. The switching strategies for the maneuver/reference selection will be defined in some rules based on the current situation from perception. When the trajectory for the maneuver is generated by PTG, the first step of the control sequence will be sent to the lower-level controllers of the continuous vehicle dynamics for execution via the available lower-level vehicle dynamics controllers (VDC), whose discussion is omitted here. The reader is referred to [30] and other standard references for this topic.

2.3.1 Assigner/Maneuver Status

As mentioned above, the assigner module could consist of several finite state machines (FSMs), based on the scenario the ACV is in, a relevant FSM will be chosen. In each FSM, the ACV can switch among different maneuvers each of which have associated setups of the MPC in the PTG. These setup actions may include:

- Filtering the nearby vehicles or moving objects as target obstacles.
- Selecting the reference states or state constraints for the vehicle to obey.

- Tuning the weighting matrices in the objective functions or even change the formation of the objective function for the MPC

The MPC setup guides the PTG to complete a specific maneuver like following the front vehicle or leading the rear vehicle within a safe gap in the longitudinal direction; keeping a lane or changing a lane laterally; or controlling the vehicle in both directions. A typical case of the vehicle states for highway maneuvers are described by the FSMs depicted in Figure 2-2. The transitions between different maneuvers are determined by the switching rules shown in Table 2-1.

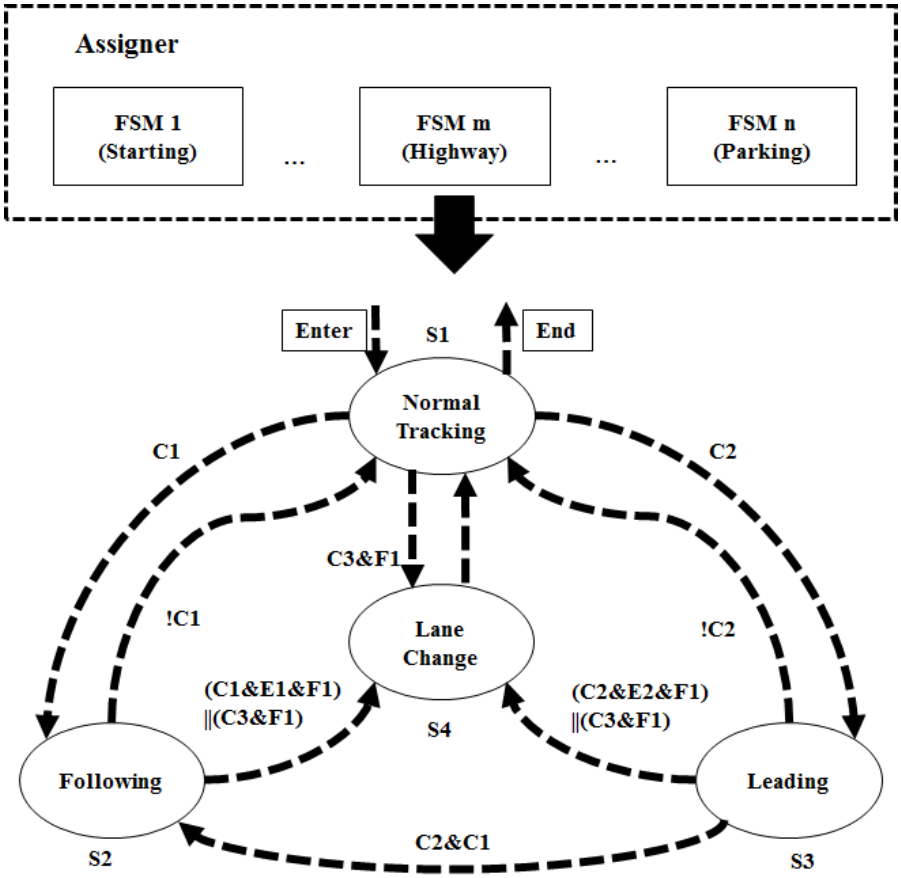


Figure 2-2: The FSM for basic highway maneuvers in the assigner

In Table 2-1, v_f and v_r represent the speeds of the detected front vehicle and rear vehicle, respectively, as observed from the autonomously controlled vehicle (ACV) in the same lane. v_t is the speed of the ACV. v_{lcl} and v_{lch} are customizable lower and higher bounds of satisfactory speeds that may be selected by occupants of the ACV. If these speed ranges are violated, i.e. $v_t > v_{lch}$ or $v_t < v_{lcl}$, a lane change will be triggered. Furthermore, v_{lcl} and v_{lch} must not violate the hard traffic speed limits $[\underline{v}, \overline{v}]$ for the lane as $\underline{v} \leq v_{lcl} \leq v_{lch} \leq \overline{v}$.

Table 2-1: Switching rules for the transition conditions

Condition	Rules	Assignment
C1	$v_f \leq v_t$	$v_{t,r} = v_f$, otherwise $v_{t,r} = v_{ref}$
C2	$v_r \geq v_t$	$v_{t,r} = v_r$, otherwise $v_{t,r} = v_{ref}$
C3	Merge or Exit Required	Potentially $y_{e,r} = y_{e,tl}$, depends on the availability of the target lane
E1	$v_t < v_{lcl}$	
E2	$v_t > v_{lch}$	
F1	Lane Change is allowed	

$v_{t,r}$ and $y_{e,r}$ are the reference speed and lateral position error that will be assigned to the MPC. The complementary set of "Lane Change is allowed" in **F1** is defined by:

$$\left| s - s_{o_{tl}} \right| < d_s \bigcup v_{t,r_{tl}} \notin \left[v_{lcl}, v_{lch} \right] \tag{2.1}$$

$s_{o_{tl}}$ is the arc length of the object vehicle at the target lane. d_s is a safety headway distance between the ACV and the preceding OV. $v_{t,r_{tl}}$ is the reference speed of the target lane after checking the availability of the lane.

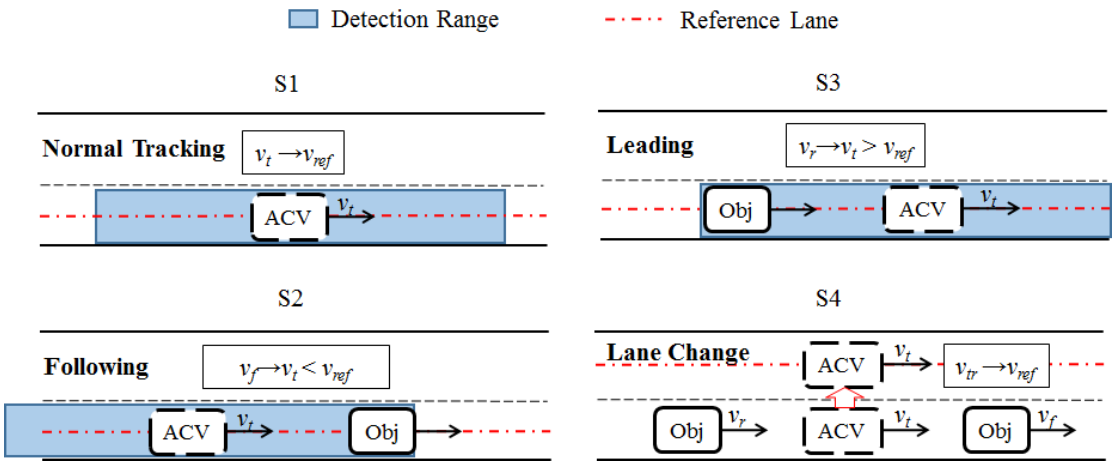


Figure 2-3: Highway maneuver states of ACV

Figure 2-3 illustrates the maneuver states and references assignment. Here v_{ref} is the reference speed for ACV and $v_{ref} = (v_{lcl} + v_{lch})/2$. The red dash dot line shows the reference lane needs to be followed. The blue rectangle illustrates the sensing range of the distance sensors. When the front OV and rear OV are too far away (outside the sensing range) or not approaching the ACV ($v_f > v_t$ or $v_r < v_t$), the ACV is in the state of S1: Normal Tracking. At this state, ACV tracks the v_{ref} and the current reference lane. If the front OV is approaching ACV ($v_f < v_t$), to avoid collision, ACV will switch from S1 to S2: Following. When the rear OV is approaching ($v_r > v_t$), ACV will switch to S3: Leading. In these two states, ACV keeps the original lane and track the speed of the approaching vehicle (v_f or v_r). However, if v_f and v_r go beyond the speed range [v_{lcl}, v_{lch}], v_t will finally stay outside the satisfied speed range, due to the speed matching requirement.

To keep v_t satisfied the customized speed bound, once $v_{t,r}$ violates [v_{lcl}, v_{lch}], ACV will switch to S4: Lane Change, if it's allowed to make a lane change based on the lane marks and the availability of the adjacent lanes. After the lane change, the state will

automatically switch back to S1. If lane change is not allowed, ACV will keep in the original states even though v_t violates the limit speed. For instance, if front OV stops, ACV will stop behind the front OV. In addition, ACV could also switch to S4 if it tends to merge in or leave the highway.

2.3.2 Vehicle Models for Predictive Trajectory Guidance

A 2D curvilinear particle motion model for ACV motion description in motion planning is used here. The reference path defined in the Frenet frame as well as the vehicle states are shown in Figure 2-4. The motion of the particle/vehicle with respect to the local reference path (lane centerline) is given by the angular alignment error ψ_e and lateral error y_e. The following equations summarize the resulting nonlinear dynamics model describing the motion of the ACV:

$$\dot{v}_t = a_t \tag{2.2}$$

$$\dot{\psi}_e = \dot{\psi}_p - v_t \cos(\psi_e)\left(\frac{\kappa(s)}{1 - y_e\kappa(s)}\right) \tag{2.3}$$

$$\dot{s} = v_t \cos(\psi_e)\left(\frac{1}{1 - y_e\kappa(s)}\right) \tag{2.4}$$

$$\dot{y}_e = v_t \sin(\psi_e) \tag{2.5}$$

where v_t is the forwarding speed of the particle vehicle, which is controlled by acceleration a_t, s is arc length of the ACV according to the reference path coordinate. The reference path is defined by its curvature $\kappa(s)$ as a function of arc length s and is assumed

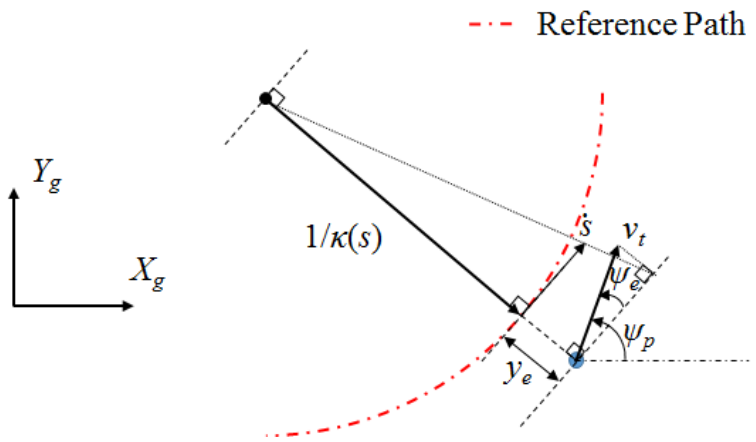


Figure 2-4: Particle motion description for the vehicle

to be known based on the map information. $\dot{\psi}_p$ is the yaw rate that used to control the angular error ψ_e. Therefore, the control inputs for the particle dynamics involve a_t and $\dot{\psi}_p$.

In additional, assuming the (lower-level) closed-loop vehicle dynamics exhibits a first-order lag behavior, the generation of control inputs to the particle dynamics can be approximate by a first order dynamics system given by:

$$\dot{a}_t = \frac{\left(a_{t,d} - a_t\right)}{T_{a_t}} \tag{2.6}$$

$$\ddot{\psi}_p = \frac{\left(v_t \kappa(\mathrm{s}) + \Delta\dot{\psi}_{p,d} - \dot{\psi}_p\right)}{T_{\psi_p}} \tag{2.7}$$

where T_{at}, $T_{\psi p}$ are the time-constants of the first-order approximation of the longitudinal and lateral dynamics of the vehicle (masked by available VDC). Therefore, the desired acceleration $a_{t,d}$ and the desired deviation from the reference yaw rate $\Delta\dot{\psi}_{p,d}$ are treated as the final inputs used to control the particle along the reference path [43].

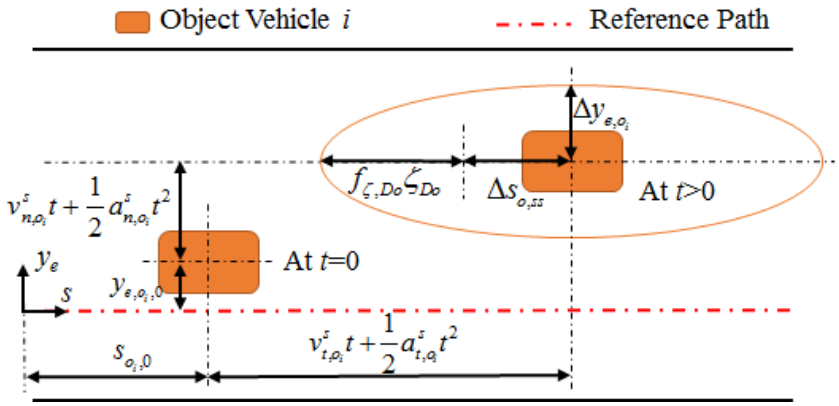


Figure 2-5: Object vehicle motion and collision avoidance constraint definition in reference path frame

As for the object vehicle (OV) shown in Figure 2-5, given the current measurement of the longitudinal velocity $v_t^s{}_{,oi}$ longitudinal acceleration $a_t^s{}_{,oi}$, lateral velocity $v_n^s{}_{,oi}$ and lateral acceleration $a_n^s{}_{,oi}$ as well as the position $s_{oi,0}$, $y_{e,oi,0}$ of the i^{th} OV resolved in the path coordinate, one can predict the future position of object i at time t in the prediction horizon using:

$$s_{o_i} = s_{o_i,0} + v^s_{t,o_i}t + \frac{1}{2}a^s_{t,o_i}t^2 \tag{2.8}$$

$$y_{e,o_i} = y_{e,o_i,0} + v^s_{n,o_i}t + \frac{1}{2}a^s_{n,o_i}t^2 \tag{2.9}$$

Here, the $a_t^s{}_{,oi}$ and $a_n^s{}_{,oi}$ are held constant for the prediction horizon, but are to be renewed at each MPC update. These update times are set to be in the order of 100-150ms, which is sufficiently fast to overcome the error due to this assumption of constant accelerations.

2.3.3 Constraints for Predictive Trajectory Guidance

As shown in Figure 2-5, the constraint to keep a safe distance between the ACV and any nearby object *i* is modeled by the elliptic inequality:

$$\left(\frac{y_e - y_{e,o_i}}{\Delta y_{e,o_i}}\right)^2 + \left(\frac{s - s_{o_i}}{\Delta s_{o,ss} + f_{\zeta,Do}\zeta_{Do}}\right)^2 \geq 1 \tag{2.10}$$

ζ_{Do} is a slack variable that allows the solver to find a feasible solution in emergency situations with auxiliary dynamics given by:

$$\dot{\zeta}_{Do} = u_{\zeta_{Do}} \tag{2.11}$$

$f_{\zeta,Do}$ is an optional tuning parameter (has a unit of time). $\Delta y_{e,oi}$ and $\Delta s_{o,ss}$ are calculated by incorporating the geometry (length and width) of the objects and the ACV. These are assumed available from sensing and/or V2V communication. The longitudinal safe distance d_s is defined by the major axis:

$$d_s = \Delta s_{o,ss} + f_{\zeta,Do}\zeta_{Do} \tag{2.12}$$

The control input must also be limited to the physical constraint of the acceleration according to the friction ellipse of a real vehicle's tire/road contact:

$$\left(v_t\left(\kappa(s)v_t + \Delta\dot{\psi}_{p,r}\right)/a_{n,gg}\right)^2 + \left(a_{t,d}\right)^2 \leq \left(\mu_H g - \zeta_{gg}\right)^2 \tag{2.13}$$

Here, μ_H is the limiting tire-road friction coefficient, g is the gravitational constant. $a_{n,gg} \in [0,1]$ is the scaling of the ellipse for lateral acceleration. The slack variable ζ_{gg} enables

the formulation of the limit value of the combined accelerations as a soft constraint with auxiliary dynamics defined by:

$$\dot{\zeta}_{gg} = u_{\zeta_{gg}} \tag{2.14}$$

Other state constraints like lane boundaries, speed limits and the minimum turning radius, etc. are also considered; for complete details, please refer to [31].

2.3.4 MPC Formulation for the Predictive Trajectory Guidance

The objective function of the MPC weighs the reference tracking error and control efforts as follow:

$$\min_{u_k} \sum_{k=1}^{N_p-1} \left(\left\| y_k - r \right\|_P^2 + \left\| u_k \right\|_R^2 \right) + \left\| y_{N_p} - r \right\|_{P_t}^2 \tag{2.15}$$

$$\text{subject to :} \qquad \dot{x} = f\left(x,u\right),\ u \in U,\ x \in X \tag{2.16}$$

$$x(0) = x_0 \tag{2.17}$$

$$0 \le c\left(x,u\right) \tag{2.18}$$

$$where: x = \begin{bmatrix} v_t & y_e & \psi_e & s & a_t & \dot{\psi}_p & \zeta_{gg} & \zeta_{Do} \end{bmatrix}^T \tag{2.19}$$

$$u = \begin{bmatrix} a_{t,d} & \Delta\dot{\psi}_{p,d} & u_{\zeta_{Do}} & u_{\zeta_{gg}} \end{bmatrix} \tag{2.20}$$

$$y = \begin{bmatrix} v_t & y_e & \zeta_{gg} & \zeta_{Do} - v_t \end{bmatrix}^T \tag{2.21}$$

$$r = \begin{bmatrix} v_{t,r} & y_{e,r} & \zeta_{gg,r} & e_{\zeta_{do},r} \end{bmatrix}^T \tag{2.22}$$

Here, $k \in (0, 1, \ldots, N_p)$ is the discretized prediction step number, where N_p is the prediction step length. The prediction horizon H_p is defined by $H_p = N_p \Delta T$ and ΔT is the sample and

update time of the MPC. x covers all the state variables of the ACV particle motion model given by (2.2)~(2.7) and the slack variables in (2.11) and (2.14). X represents the state-space for x. x_0 denotes the current/initial state (measured). The system outputs, namely the speed v_t and lateral position y_e of the ACV as well as the slack variable output ζ_{Do}, ζ_{gg} are grouped in vector y. U denotes the admissible set for input u, which includes the input to ACV motion model and selection variables. P and R are the weighting matrices for tracking error and control efforts. P_t is used to weight the terminal cost in tracking error. All the inequality constraints like (2.10), (2.13) and other constraints are included in (2.18).

The MPC formulated above is implemented using the ACADO Toolkit and accompanying Code Generation Tool from [44] [33] [45]. Therein, a sequential-quadratic programming (SQP) algorithm generates a quadratic approximation of the nonlinear problem and solves it with an online active-set QP solver from the open-source library qpOASES. Some restrictions and facilities in the tool influenced the formulation of the MPC model adopted here more than others. For example, instead of formulating the obstacles to avoid as polygons, each of which are in turn defined by intersections and unions of regions defined by linear constraints, the elliptic constraint formulation allows one to use few analytical functions to describe each obstacle. The drawback is this could lead to some conservatism in describing the avoidance area. In our analysis, we used the multiple shooting horizon discretization option and a 4th order explicit Runge-Kutta integrator. For details on the toolbox, the interested reader is referred to [45] and the references therein.

While it is not possible to give upper bounds on the number of iterations needed to arrive at the optimal solution of the nonlinear optimization problem with active set QP solvers, some computational scalability bounds can be given for a single QP iteration. Following discussions in [44], it can be shown that the computational complexity of one QP iteration of the active set solver selected is $O(N_x^3+N_u^2+N_p^2+(N_uN_p)^2+N_uN_pN_c)$, where N_x is the number of states, N_u is the number of control inputs, N_p the length of the prediction horizon, N_c is the number of constraints. Thus, the additional control inputs, states and constraints we added for the slack variables lead to an increase of the computational load. However, these additions are deemed acceptable as they make the problem more tractable.

2.4 Simulation Results and Conclusions

2.4.1 Simulation Results

In this section, several cases are simulated to illustrate the performance of the proposed control framework in autonomous driving. The PTG uses the motion model and MPC formulation detailed in the previous sections implemented with the ACADO code generated solver described above. Unless specified otherwise, the following MPC settings are used: the prediction model is discretized with ΔT = 0.15s and N_p= 40, thus H_p= 6s the MPC is updated with T_{mpc} = 0.05s (This is higher than all execution times tested). The other parameters chosen are listed in Table 2-2.

Table 2-2: MPC parameters

Parameter	Value	Parameter	Value
v_{lch} [m/s]	28	μ_H	1
v_{lcl} [m/s]	23	P_{v_t}	1.1
T_{at} [s]	13.3	P_{y_e}	3
$T_{\dot{\psi}_p}$ [s]	5	$P_{\zeta_{gg}}$	20
$\Delta y_{e,o_i}$ [m]	5.3	$P_{\zeta_{do}}$	20
$f_{\zeta,Do}$	1	R_{a_t}	20
$\Delta s_{o,ss}$ [m]	2.3	$R_{\Delta\dot{\psi}_p}$	250

First, we consider a scenario where the ACV needs to follow only one reference path through an intersection with a left turn in the presence of two other vehicles (object vehicles), see Figure 2-6. The ACV is restricted with ±0.75m tolerance along its reference path. The traffic light in front of it turns red at t= 5s when s = 90 m and it turns green at t= 20s. Suppose object vehicle 2 (OV2) violets the red light it faces at t= 20s and keeps going through the intersection from left to the right. Suppose also that Object vehicle 1 (OV1, driven by an attentive driver) yields to OV2 and starts moving forward at t= 27s. The predicted and updated plans for the ACV are shown in Figure 2-6. Before t= 5s, PTG plans to slow down the ACV and make a left turn and go through the intersection, therefore, the lateral acceleration a_n magnitude increases in the planning/prediction. After detecting the red light, the plan changes to a further deceleration and finally, a complete stop in front of the traffic light, constrained by $\bar{s}$ = 90m. When the light turns green, when the arc length

restriction is softened to allow the ACV to accelerate. However, as OV2 doesn't obey its traffic light, PTG plans to yield to it to avoid collision and wait till OV2 has crossed the intersection. Then, the PTG guides the ACV to reduce its acceleration to let OV1 pass the intersection.

Secondly, we consider a series of highway scenarios with two lanes. The initial conditions for the straight lane scenarios are shown in Figure 2-7(top). The forward

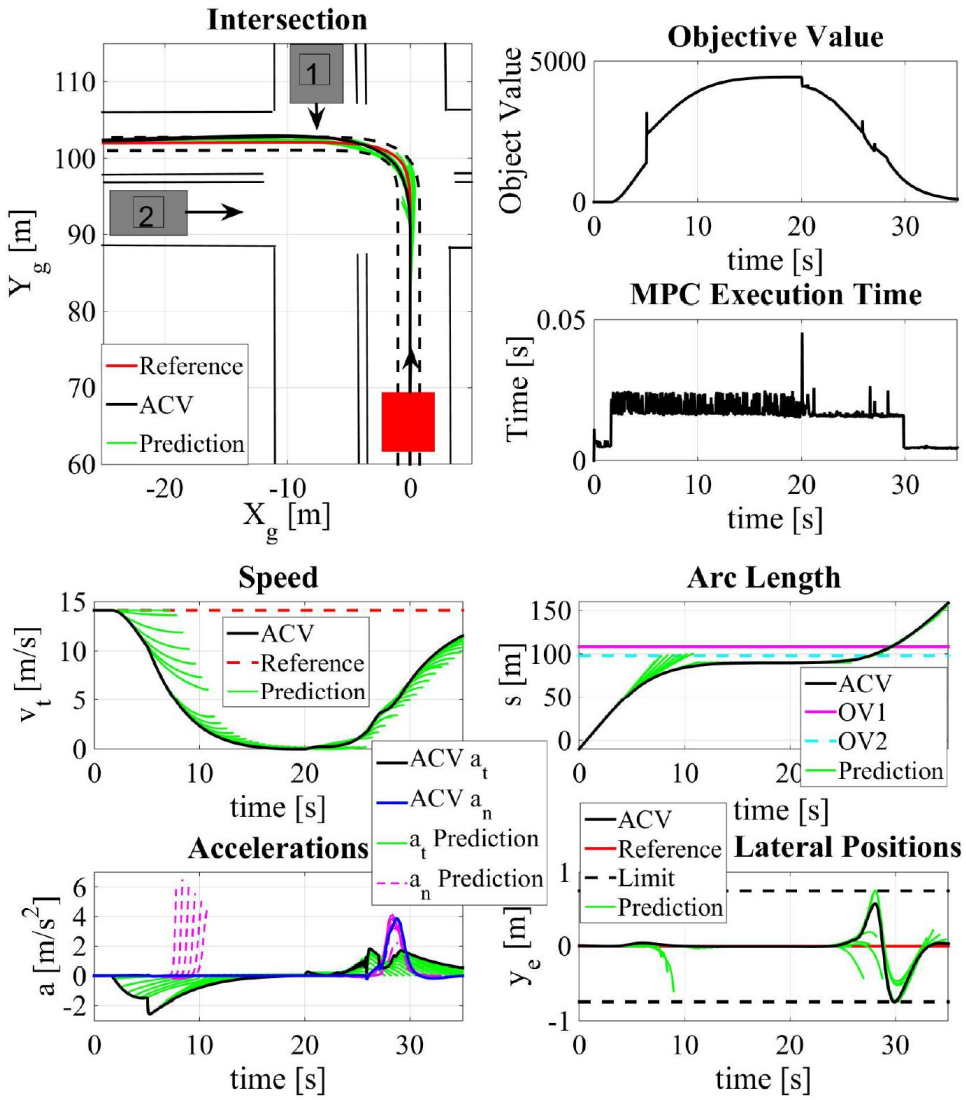


Figure 2-6: Results for an intersection scenario

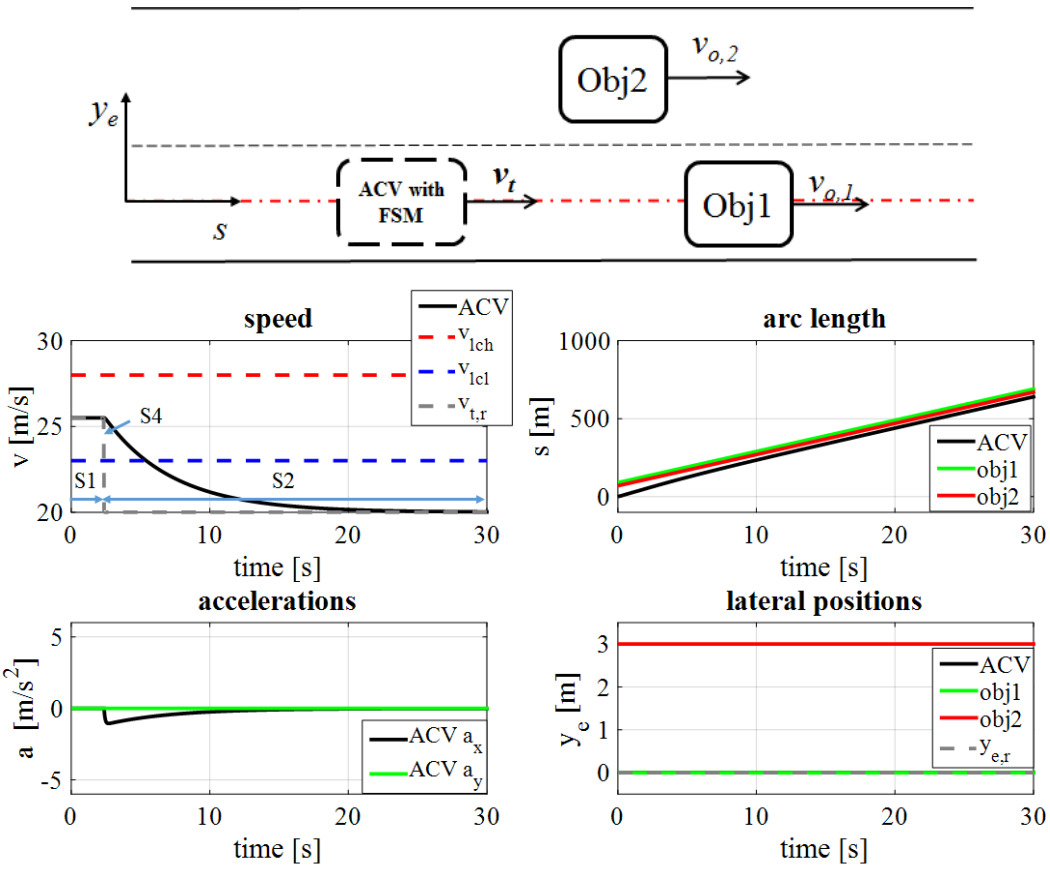


Figure 2-7: Initial condition(top) and results(bottom) for an straight highway scenario with FSM assigner invoked

directions of the two lanes are the same. Figure 2-7 and Figure 2-8 show the results for an unsuccessful passing of the ACV due to the narrow space available for ACV to pass. We consider cases with (Figure 2-7) and without (Figure 2-8) the FSM assigner invoked. If the assigner is not invoked, velocity reference $v_{t,r}$ and lateral distance error reference $y_{e,r}$ will not change in (2.22), thus only normal tracking maneuver (S1), which can nominally avoid obstacles, will be chosen. In this Scenario, as Obj1 and Obj2 share the same speed and stay close to each other, their elliptical regions overlap with each other and block the road, as shown in Figure 2-8 (top), which results in a local minimum for the cost function at the

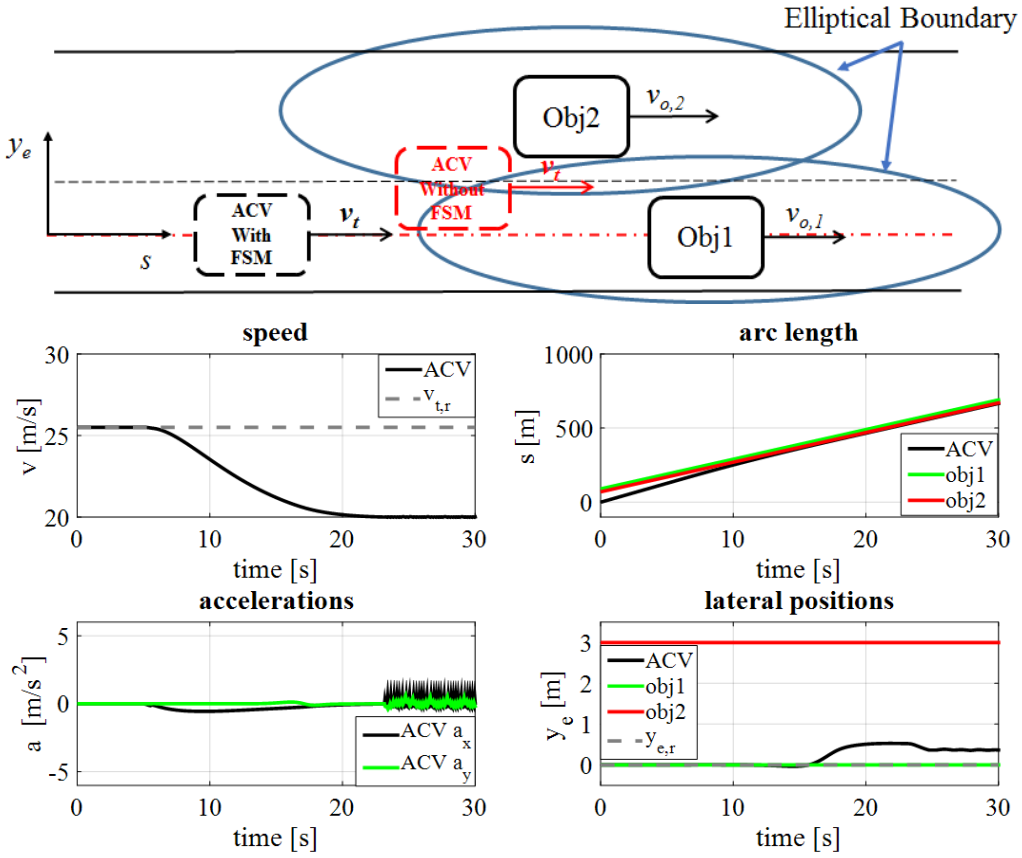


Figure 2-8: Final condition(top) and results(bottom) for an straight highway scenario without FSM assigner invoked

intersection of the two boundaries. As the MPC tries to minimize the cost function in (2.15), the ACV without the FSM assigner will try to overtake Obj 1 and then be trapped at this local minimum point and deviate from the original centerline, as depicted in Figure 2-8. This behavior is undesired because the ACV would (nearly) occupy two lanes simultaneously. This situation can be avoided by assigning a new reference speed to the MPC objective with the FSM assigner. In this case, the ACV with FSM (Figure 2-7) will keep following the front vehicle starting at t=1s without any lateral deviation.

A more complex scenario where the ACV tries to overtake two slow vehicles on an S-shaped highway is shown in Figure 2-9. In this case, the two vehicles/objects are initially

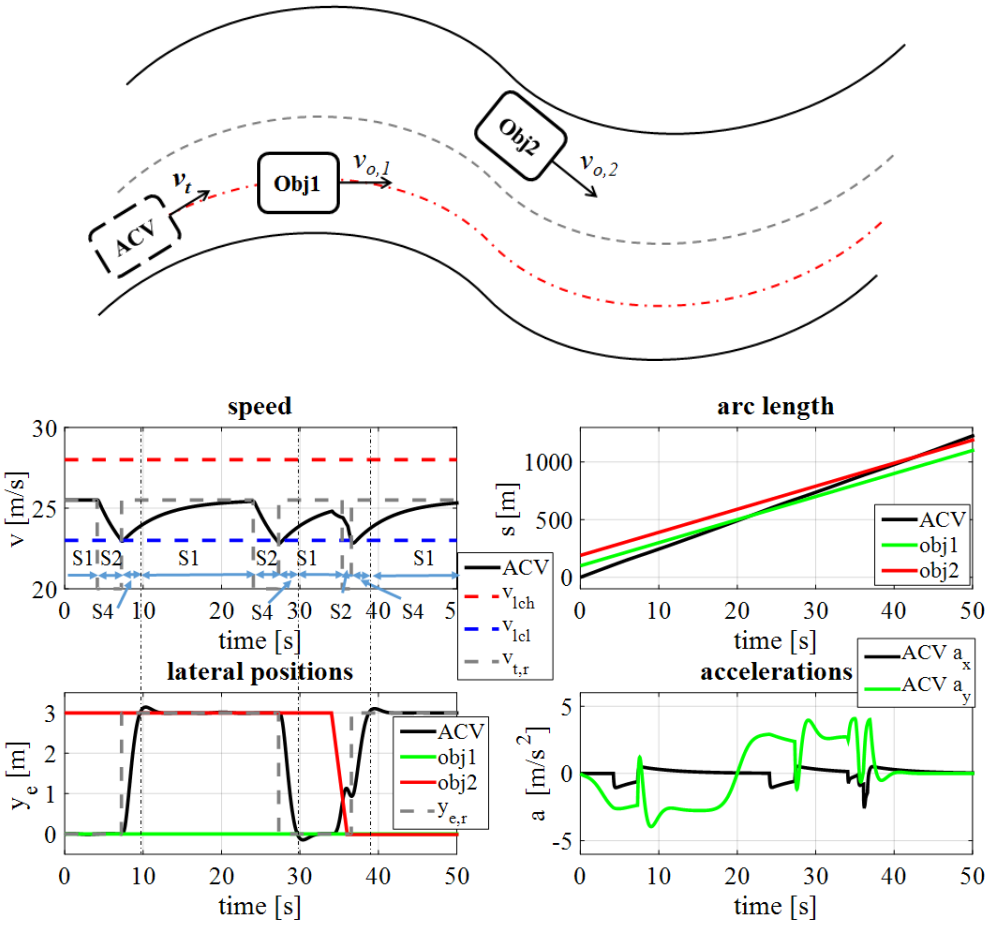


Figure 2-9: Initial condition(top) and results(bottom) for an S-shape highway scenario

in front of the ACV in different lanes. The ACV will change lane to overtake Obj1 and then repeat the maneuver to pass Obj2. A sudden lane change of Obj2 is pre-defined before the ACV tries to pass it. Therefore, the ACV has to avoid Obj2 twice. Figure 2-9 shows the results for this case. In this complex scenario, the ACV exhibits higher lateral acceleration a_y levels compared with the straight lane scenarios discussed above. The procedure of passing Obj1 is similar as in previous cases. In addition, the system makes good plans for emergency handling when dealing with Obj2 as its sudden lane change is initiated. The ACV slows down and steers in advance of Obj2 reaching the right lane to

avoid entering the danger area (S1).Then it tries to follow Obj2 (S2) before the satisfactory speed v_{lcl} is violated, which triggers the final lane change (S4). Finally the ACV overtakes Obj2 in the left lane.

2.4.2 Conclusions

In this chapter, a hybrid predictive trajectory guidance framework is outlined for autonomous vehicle control (ACV). In particular, an assigner module is detailed with several maneuver states to guide an MPC-based predictive trajectory guidance (PTG) module. The maneuver states are organized through a finite state machine (FSM) with specified transition conditions. Each maneuver state is related to a setup of the MPC references, hard constraints or weighting matrices, which will be assigned to the PTG for execution if the related maneuver state is chosen by the assigner. The PTG is based on a particle motion model for the vehicle dynamics and the path expressed in a curvilinear coordinate frame. The control inputs are generated by satisfying constraints describing dynamic public traffic, and vehicle-road friction limits. To illustrate the performance of the framework, a finite state machine of intersection and highway maneuvers are designed and simulated. The ACV under this hybrid controller design showed good performance and proper behaviors in various scenarios.

However, the maneuver selection is neither optimized nor predictively conducted, therefore the switching time from one maneuver to another might not be the best. Also the switch transient process could be uncomfortable with jerk. Furthermore, the uncertainties in the state observation is not considered. In next chapter, a predictive maneuver planning framework with observer design is introduced to address these issues.

CHAPTER 3

PREDICTIVE MANEUVER PLANNING AND CONTROL FRAMEWORK

3.1 Abstract

In this chapter, a predictive planner is proposed that combines maneuver planning and trajectory planning in a short future horizon in the presence of uncertainties. Based on the predicted likely motion of the autonomous vehicle and other object vehicles, a predictive reference speed pre-planning is operated for each lane at each time step of the prediction horizon. Then, an optimization problem is configured that computes safe, sub-optimal plans for the trajectories of both the states (and inputs) and maneuver references for the prediction horizon. While a first formulation of this results in a mixed-integer nonlinear programming (MINP) problem, it is shown that a relaxation can be adopted that reduces the computational complexity of the optimization solver to a low-order polynomial time nonlinear program which can be solved efficiently at real time. The proposed predictive maneuver planning method is illustrated through simulation of a series of multi-lane highway scenarios and compared with a one-maneuver planning approach and an adaptive cruise control approach.

3.2 Introduction

Autonomous driving is a promising technology for improving the safety, efficiency and environmental impact of on-road transportation systems. Despite the existence of elegant algorithms [46] for route or global path planning from position A to B, the task of guiding an autonomous vehicle to rapidly and systematically accommodate the plethora of

changing constraints for local motion planning in public traffic is a challenge problem. These constraints arise from tire/road friction conditions, avoiding stationary and moving obstacles, obeying the traffic rules, signals and so on. One of the core problems is designing robust and computationally efficient trajectory planning algorithms that can generate the appropriate vehicle maneuvers as well as the constituent motion trajectories while considering the differential vehicle dynamics of the controlled vehicle and the listed constraints in public traffic with measurement noise and other uncertainties. Plenty of methods have been proposed to deal with this problem, as also summarized in [47] and [48]. They roughly fall into three groups: sampling-based planning methods, path-velocity decomposition methods and numerical optimization methods.

Sampling-based planning methods are popular methods for trajectory guidance of robotic vehicles. The methods discretize/sample the state space of the motion into a library of quantized motion primitives/lattices [49], obtained from numerically solving the steady state or transient vehicle dynamic motion models. As each primitive/lattice indicates a maneuver, the methods are also called maneuver-based planning methods [18]. Then, efficient heuristics for deterministic or stochastic searching, such as the A* algorithm [50] or RRT* based algorithm [51], can be applied in real-time to construct a periodic planning law from the library, ensuring some robustness and safety in a disturbance environment. However, the completeness and optimality of these methods depends strongly on the resolution of the library. The complexity of finding the best trajectory increases with the resolution of the library. Also, the resulting non-continuous trajectory induces jerky, uncomfortable motions.

Path-velocity decomposition methods decompose the planning work into two sub-problems: local path planning and path-tracking. Graph-search based method like Dijkstra's Algorithm [52] and A* algorithm [53] or interpolating curves like clothoid curves [54], polynomial curves [55] and Bezier curves [56] are used in the local path planner design to generate the way points in the 2D configuration space. Then, a closed-loop controller is applied to track the path while satisfying the constraints in work space. However, as the planned path is not often given as a function of time, collision-free motion is not guaranteed by following the path. Therefore, the robustness and safety of the decomposition methods highly depend on the quality of the path-tracking controller.

On the other hand, the numerical optimization methods find the best trajectory by solving constrained optimization problems. These methods can naturally handle multiple constraints and uncertainties but they suffer from the computational burden of optimizing the motion state over a future horizon from a current time step. Therefore, in practical applications, these methods usually follow a receding horizon pattern with a limited horizon length in a scheme also known as model predictive control (MPC). These use fast real-time solvers [33] [32] to periodically solve the optimization problems, where only a first section (step) of the input trajectory is executed and the process is repeated in receding prediction horizons. MPC, which initially was applied to modeling human-driver like control in various traffic situations [57], it now appears in many works as a reactive planner for autonomous and semi-autonomous vehicle control [58, 59, 31]. To apply MPC for trajectory planning requires the knowledge of global route waypoints as references to follow. For the on-road scenario, the centerline of each lane from the perception or map

information [60, 61] can be used as the reference path. For off-road scenarios, this method can be incorporated in the path tracking level the path-velocity decomposition methods [62]. In addition, specific terminal costs and constraints could be designed to circumvent limitations of robustness and stability that arise from the use of finite horizons [63, 64].

In MPC-based trajectory planning, the expected states of the autonomously controlled vehicle (ACV or ego vehicle) and other object vehicles (OVs) are model-predicted for the duration of the prediction horizon based on the current measurements. This allows the ACV to assess the risk of having a collision with other OVs and then to determine a collision-free trajectory. Different models used for motion prediction of OVs are summarized in [65], including physics-based models, maneuver-based models and interaction-aware models. Physics-based models [66, 67] simply assume constant velocity or constant accelerations and thus they can only be used in motion prediction for a short term (less than 1 second). Maneuver-based models [68, 69] predict the motion based on the estimation of maneuver intentions. Interaction-aware models [70, 71] also consider the inter-dependencies between the individual vehicles' maneuvers. The latter two models allow longer-term prediction compared to physics-based models. The interaction-aware model is more reliable than maneuver-based models, but it's also much-more computationally expensive, difficult to fully characterize, and is not compatible for real-time risk assessment [65]. Maneuver-based models remain the viable options for real-time long-term motion prediction (more than a second).

The planning problem naturally involves uncertainties due to modeling error, sensor imperfections or environmental disturbances, as summarized in [72]. In prediction

of the motion of the OVs for risk assessment, the uncertainties can be handled by either using reachability analysis [73, 74, 75] (estimating the propagation of the uncertainty bounds) or using stochastic reachability analysis [76, 77] (estimating the propagation of the uncertainty distribution). In the reachability analysis case, the worst case of the uncertainty is considered thus leading to a very conservative solution for the planning problem. However, for stochastic reachability analysis, the reachable set as well as the risk of collision can be assessed by probabilities. If the state uncertainties are Gaussian distributed, the stochastic reachability analysis can be implemented via filtering techniques, e.g., Kalman filter (KF) series [78, 79] for motion prediction of one maneuver and Interactive Multi Model (IMM) KF or Switching KF [80] for different possible maneuvers. Therein, the computational process for solving for the collision free trajectory of the ACV in the MPC with filtering techniques is similar to applying stochastic reachability analysis applied to find a fail-safe trajectory [77].

Also, due to the sub-optimality caused by the nonconvex configuration space, and the warm start strategy usually applied, an ACV with static or pre-configured optimization setups cannot handle the complex scenario and could get trapped at undesired local minima. Therefore in last chapter, a predictive control framework which can switch from a combination of rule-based discrete maneuver decisions is applied. With these rule-based decisions, the planned trajectories are can switch among different maneuver to accommodate complex traffic and be forced out of undesired local minima. To improve the optimality of the maneuver decisions, the main contribution in this chapter, we configure these decisions of the assigner module to be selected optimally and predictively

so as to change the maneuver states as frequently as necessary for the prevailing driving scenario. However, the optimization of such a problem involves solving for the discrete maneuvers as well as the local motion trajectories [81]. This leads to a mixed-integer nonlinear programming problem (MINP) [35], which requires specific heuristic-based solvers to find a solution. To overcome this problem, another contribution of this chapter is the introduction of relaxation constraints to transform the MINP into a regular nonlinear programming which can be efficiently solved in real time. We also compare the computational complexity of the naive MINP formulation with the proposed relaxation technique for the specific active set solver adopted. Here, we explicitly include uncertainties in the estimation and prediction of the states of the ACV and of the OVs as well as in determining the evolution of the tightened MPC constraints to explicitly accommodate uncertainties. The performance of the proposed predictive maneuver planning is illustrated via the simulation results in comparison with the one maneuver selection approach and a regular adaptive cruise control (ACC) approach under several complex scenarios on a highway.

3.3 Framework Design

Figure 3-1 shows a schematic of the proposed predictive maneuver planning and control framework for an autonomous road vehicle in a uncertained public traffic environment. At the top the assigner module integrates/fuses the information from the environment perception module (lane detection, traffic sign and signals, object tracking…), the navigator/path planner module (route navigator for on-road situation and path planner for off-road situation), the vehicle dynamics sensing/estimation modules (planer and yaw

motion of the ACV) and motion prediction module (both ACV and tracked OVs). Here, we assume the states of the ACV x_a and OV x_o are fully observed/estimated. The fused information will be provided to some pre-defined finite state machines (FSMs)/maneuver automatons and then the decisions on configurations, e.g., parameters (p), references (r), and constraints (c), for the MPC formulation will be made and assigned to the predictive trajectory guidance (PTG) module. Further description of these pieces of information can be found in [31] and will be also briefly mentioned in Section III.

The maneuver automatons/FSMs designed and stored in the assigner module are scenario-based. Once the possible maneuvers for a scenario are captured, like cruising, following, leading and lane change for highway scenario [82] (see Figure 3-2), or going

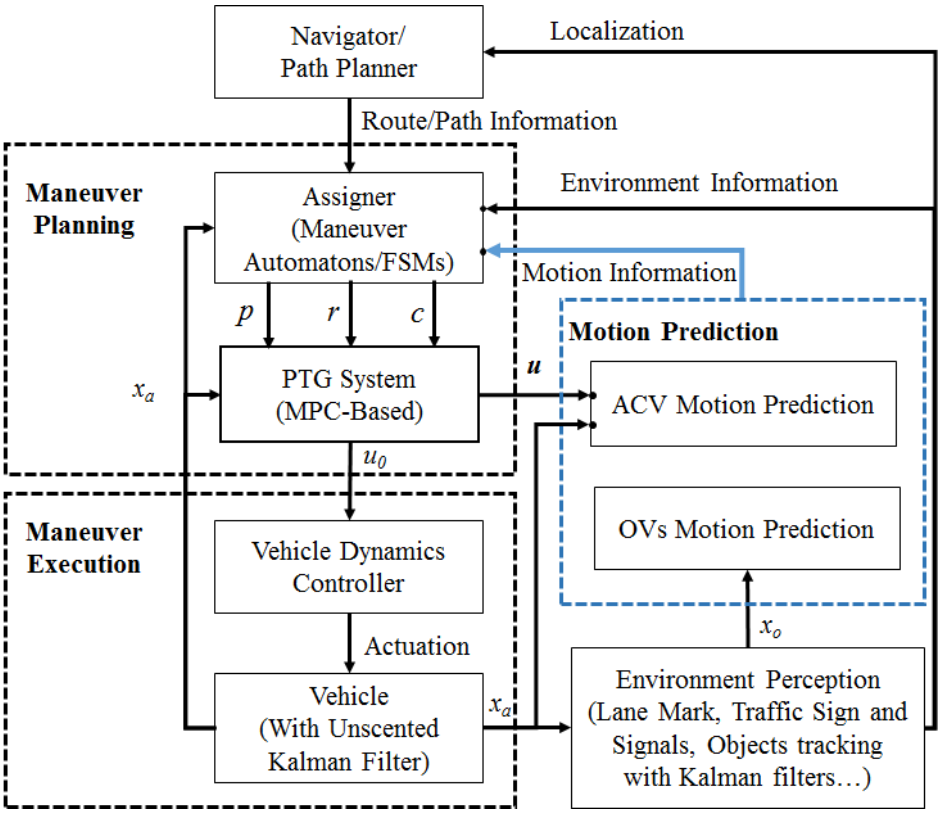


Figure 3-1: Hierarchical predictive maneuver planning and control framework

left/right/straight and stop for intersection scenario [83], the relevant FSM can be easily extended for the same scenario or to other scenarios [38]. The candidate maneuvers in the FSMs are related to their own references, e.g. desired speed and lane. At each prediction interval of the MPC, the references are pre-selected according to the predicted motion of the ACV and the surrounding OVs via filtering techniques. Then, the optimized maneuver sequence as well as the optimized relevant control output trajectory $\boldsymbol{u}$ for the whole horizon are solved simultaneously by the PTG system, according to the objective function and constraints to be detailed later. When the maneuver planning is done, the first interval of $\boldsymbol{u}$, i.e., u_0 is sent to the lower-level controllers of the continuous vehicle dynamics for execution via the available lower-level vehicle dynamics controllers (VDC), whose discussion is omitted here. The reader is referred to [30] [31] and other standard references for this topic.

3.3.1 Vehicle Models and Filtering Design

The similar particle model for motion description of ACV and OV used in Chapter 2 (2.2)~(2.9) are used here with the consideration of uncertainties. For convenience, we shall present here the continuous time models of the state dynamics even if computations are ultimately to be done in discrete time form. Adding process uncertainties (random disturbances and uncertainties) and measurement noise, the nonlinear dynamic model describing the motion of the ACV can be written as:

$$\begin{bmatrix} \dot{v}_t \\ \dot{\psi}_e \\ \dot{s} \\ \dot{y}_e \\ \dot{a}_t \\ \ddot{\psi}_p \end{bmatrix} = \begin{bmatrix} a_t \\ \dot{\psi}_p - v_t \cos(\psi_e)\kappa(s)/[1-y_e\kappa(s)] \\ v_t \cos(\psi_e)/[1-y_e\kappa(s)] \\ v_t \sin(\psi_e) \\ -a_t / T_{a_t} \\ [v_t\kappa(s) - \dot{\psi}_p]/T_{\dot{\psi}_p} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1/T_{a_t} & 0 \\ 0 & 1/T_{\dot{\psi}_p} \end{bmatrix} \begin{bmatrix} a_{t,d} \\ \Delta\dot{\psi}_{p,d} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_{a_t} \\ w_{\Delta\dot{\psi}_d} \end{bmatrix}$$

$$\begin{bmatrix} y_s \\ y_{y_e} \\ y_{a_t} \\ y_{\dot{\psi}_p} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_t \\ \psi_e \\ s \\ y_e \\ a_t \\ \dot{\psi}_p \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_s \\ v_{y_e} \\ v_{a_t} \\ v_{\dot{\psi}_p} \end{bmatrix}$$

(3.1)

The additive Gaussian process noise $w = [\, w_{a_t} \;\; w_{\Delta\dot{\psi}_d} \,]$ are used to model disturbances (e.g. wind, road, unmolded dynamics) affecting the generation of longitudinal acceleration a_t and yaw rate $\dot{\psi}_p$ of the ACV. For the system outputs, we consider that the available measurements are positions s, y_e (e.g., from GPS) and the inertial states a_t, $\dot{\psi}_p$ (e.g., from IMU) with assumed Gaussian sensor noise $v = [\, v_s \;\; v_{y_e} \;\; v_{a_t} \;\; v_{\dot{\psi}_p} \,]^T$.

For an OV, its motion is also defined in the Frenet frame as a particle. To consider its maneuver intention for better motion prediction, we assume each OV to follow closed-form dynamics that describe longitudinal motion like cruising at a specific speed or speed change and lateral motion like tacking a specific lane or lane change. One possible form is:

$$\begin{bmatrix} \dot{s}_o \\ \dot{v}^s_{t,o} \\ \dot{y}_{e,o} \\ \dot{v}^s_{n,o} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-K_{s1}}{1+K_{s2}} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -K_{y1} & -K_{y2} \end{bmatrix} \begin{bmatrix} s_o \\ v^s_{t,o} \\ y_{e,o} \\ v^s_{n,o} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{K_{s1}}{1+K_{s2}} & 0 \\ 0 & 0 \\ 0 & K_{y1} \end{bmatrix} \begin{bmatrix} v^s_{t,o,ref} \\ y_{e,o,ref} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{1+K_{s2}} & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_{\dot{s},o} \\ w_{y,o} \end{bmatrix}$$

$$\begin{bmatrix} y_{s_o} \\ y_{y_{e,o}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_o \\ v^s_{t,o} \\ y_{e,o} \\ v^s_{n,o} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{s,o} \\ v_{y,o} \end{bmatrix}$$

(3.2)

where, s_o and $y_{e,o}$ are the arc length and lateral position error of the OV; $v^s_{t,o}$ and $v^s_{n,o}$ are the tangential speed and normal speed of the OV along its reference lane. K_{s1} and K_{s2} are the proportional and integral gains of a controlled OV tracking the reference speed $v^s_{t,o,ref}$ with assumed Gaussian process noise $w_{\dot{s},o}$. K_{y1} and K_{y2} are the proportional and integral gains of a controlled OV tracking its reference lane $y_{e,o,ref}$ with assumed Gaussian process noise $w_{y,o}$. These gains can be identified from the human-driver data to emulate different driving habits, e.g. either aggressive or conservative [84]. For system outputs, only the positions s_o and $y_{e,o}$ are assumed measured with associated Gaussian sensor noises $v_{s,o}$ and $v_{y,o}$ (e.g., from on-board range sensors like radars on the ACV).

Given the nonlinear system model in (3.1), we adopt Unscented Kalman Filter (UKF) [85] to estimate the motion states of ACV in the presence of process and measurement uncertainty/noise. Given the linear dynamics motion models for the OVs (3.2), a regular KF can be used for state estimation of one maneuver (tracking a specific lane and speed). To account for other possible maneuvers of the OVs, the Interactive Multi-

Model KF algorithm [80, 86] can be applied for OV state estimation. Here, we assume the gains are well captured from the driving data for the drivers of all OVs of all maneuvers.

Given the current estimates of the ACV and OV states, one can predict the evolution of the mean and covariance of the states for the whole length of the prediction horizon of the MPC. Here, we propagate uncertainty in the predicted states (for both ACV and OV) using the filtering techniques (UKF/KF) based on (3.1) and (3.2) with the notion of the most likely measurement. This notion is based on the assumption that the future measurements in the update of the filter recursions is approximated well by the prediction. This assumption is motivated by the fact that future measurements are unavailable. Even though the updated covariance is not directly affected by the value of the measurement, as the measurement information is considered (via the only assumption that same sensors and models are to be used), the uncertainties in the likely state are reduced. It is shown in [78] that the most likely measurement will not introduce bias in the system, thus it is useful to constrain the uncertainty propagation. Finally, note that the future inputs used in the motion prediction of the ACV will be taken from the previous planning results of the MPC.

The above models for motion prediction of OVs and the ACV do not explicitly consider the interactions between vehicles, particularly those that would exist in mixed-traffic involving other human-driven vehicles. As alternatives, other motion prediction approaches such as interactive multi-model filtering, Bayesian Networks, and Hidden Markov Models trained on human-driver data are all possible options [71] [87]. While any of these approaches may be used for motion prediction and incorporated with the maneuver planning framework presented in this chapter, as we point out later, for multiple OVs in

multi-lane scenarios, the computational complexity of using even linear motion models for the OVs needs to be handled with care.

3.3.2 Maneuver Automaton

The hybrid system notion is straightforward to apply to the motion of a road vehicle, since basic maneuvers, like accelerating, cruising or decelerating in the longitudinal direction and steering to the left or right in the lateral direction can be identified from the vehicle's motion [82, 83]. For the ACV, these maneuvers can be designed via tracking

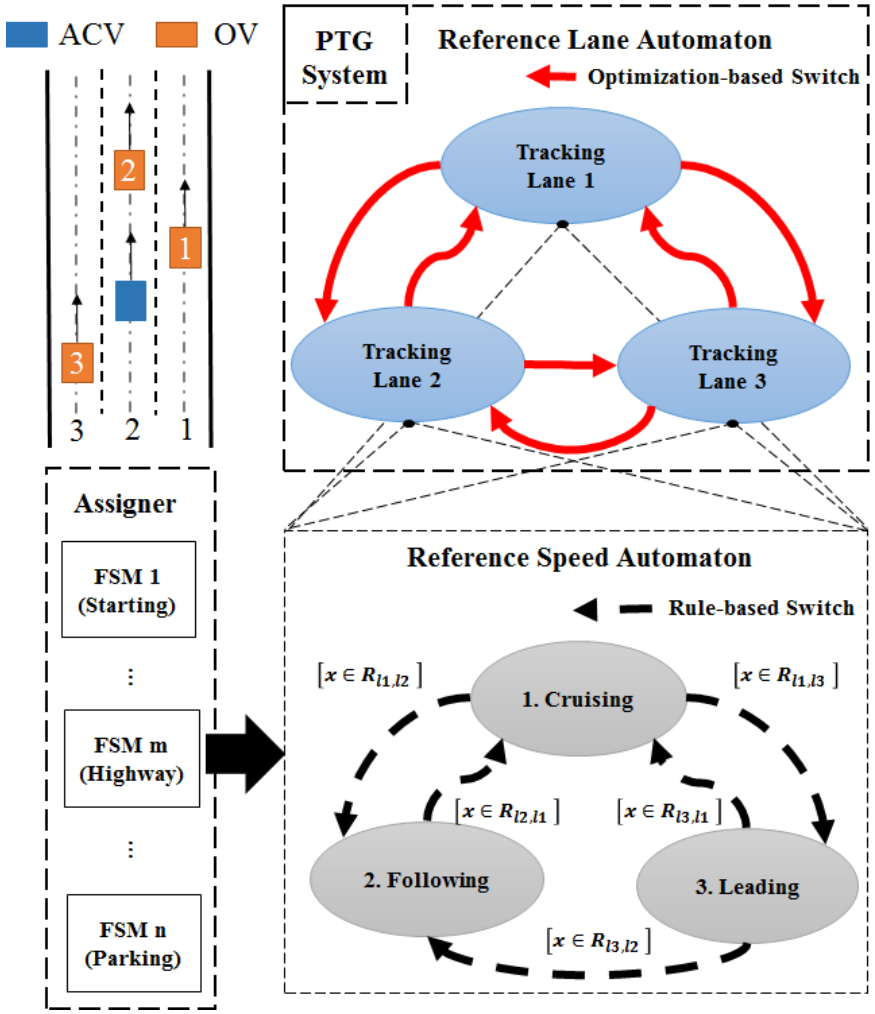


Figure 3-2: Maneuver automaton example for 3-lane highway scenario. Rule-based switch sets are denoted by *R*.

different reference speeds and reference paths/lanes, as we proposed in Chapter 2. This results in a hybrid system model involving tracking of *two-dimensional* discrete references (speed and lane) and the underlying continuous vehicle motion trajectories. To construct the maneuver planning for a prediction horizon with low complexity, the switching among the discrete maneuvers can be done hierarchically (see Figure 3-2, for example): 1) Firstly, the switching of the reference speeds assigned for each lane based on pre-defined rules (rule-based switch sets) is executed at each prediction step in the horizon. This is called rule-based switch. 2) Then, an optimization problem is solved for the whole horizon to find the optimized switching sequence for the reference lanes. This is called optimization-based switch. For different scenarios, the maneuvers and the rule-based switch sets can be specifically designed and stored in different FSMs of the assigner module, e.g., single lane, intersection, etc.

Considering the interaction of the vehicle with the surrounding dynamic environment, e.g. the traffic sign, signals and OVs, for example, when approaching a slow front OV, a normal reaction of the vehicle will be either slowing down to follow it or simply changing lane to overtake it. Those intentions can be reflected by the reference speed assignment to the ACV. Specifically, on lane l, a relevant reference speed $v_{t,r,l}$ will be assigned to the ACV to follow, depending on the detection of approaching or close by object vehicles, as shown in . The detection condition (3.3) and approaching condition (3.4) are defined by:

$$\left|\hat{s} - \hat{s}_{o_i}\right| < T_d v_{t,ref} \tag{3.3}$$

$$\left(\hat{s}-\hat{s}_{o_i}\right)\left(v_{t,ref}-\hat{v}^{s}_{t,o_i}\right)<0 \tag{3.4}$$

where T_d is the detection preview time (set from specifications of the perception module, its value should be larger than the predictive horizon H_p to prevent the ignorance of an abrupt event from the surrounding traffic for MPC). Here, and in the following, the usual hat (^) notation is used to denote the respective estimated states. The i^{th} OV occupying lane l is denoted by:

$$\hat{y}_{e,o_i} \in \left[\underline{y_{e,l}},\ \overline{y_{e,l}}\right] \tag{3.5}$$

where $y_{e,oi}$ is the lateral position of OV i in the path coordinate and the lane l is demarked by the lateral position bounds $\left[\underline{y_{e,l}},\ \overline{y_{e,l}}\right]$.

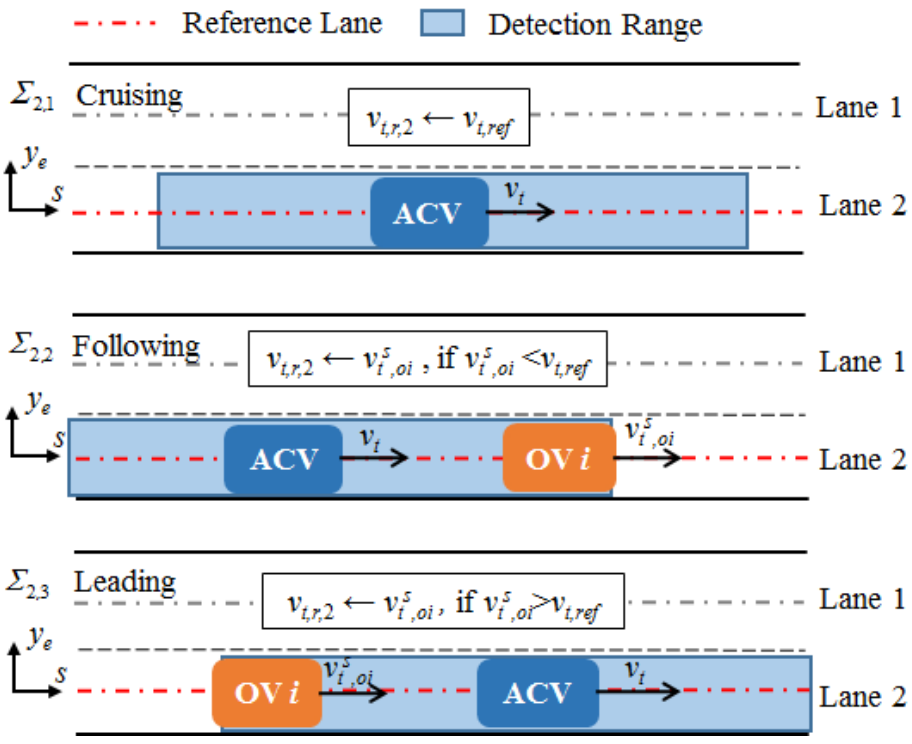


Figure 3-3: Reference speed assignment for ACV in 2 lane scenario.

Accordingly, the cruising maneuver is defined as tracking a desired cruise speed $v_{t,ref}$ on lane l (lane 2 in Figure 3-3) without detecting any approaching OVs on the same lane. The corresponding speed relation and reference speed assignment is given by:

$$v_{t,r,l} = v_{t,ref} \tag{3.6}$$

The following or leading maneuver refers to tracking the speed $v_t^s{}_{,oi}$ of a detected approaching OV on lane l in the front or rear by assigning:

$$v_{t,r,l} = \hat{v}^s_{t,o_i} \tag{3.7}$$

Remark 3-1: Given a prediction horizon of length N_p steps, the rule-based reference speed automaton is applied for every lane at every prediction interval. This means a reference speed sequence with N_p elements will be generated for each lane, thus effectively constructing a two-dimensional pre-plan of the references in both the longitudinal and lateral directions. This will be used later in the MPC to find the best sequence of reference selections for the whole horizon that minimize an objective function. Note that there is a possible loss of performance from the non-optimality of the reference speed assignment rules; but these are discrete rules done outside of the MPC; the optimization of such assignment rules is beyond the scope of this chapter.

Generally, the ACV is expected to track the desired cruise speed $v_{t,ref}$ within the acceptable speed range $[v_{t,cl}, v_{t,ch}]$ with positive speed tolerance Δv_t:

$$\begin{cases} v_{t,cl} = v_{t,ref} - \Delta v_t \\ v_{t,ch} = v_{t,ref} + \Delta v_t \end{cases} \tag{3.8}$$

However, as argued in [88], by following the optimization-based reference lane automaton introduced in the next section, the ACV can be "trapped" in one lane in a following or leading maneuver due to the formulation of MPC objective function with a lane selection variable (to be described later). In such situations, a forced lane change is necessary to help the ACV jump out of the trap. Therefore, we extend the rules used in [89] to guide the ACV to an adjacent empty lane or to one with the assigned speed closest to the desired cruise speed via reference speed adjustment. If the assigned reference speed of the ACV in the current lane l goes outside of this speed range:

$$\hat{v}_{t,r,l} \notin \left[v_{t,cl}, v_{t,ch} \right] \tag{3.9}$$

and adjacent lane(s) are unoccupied or are with assigned speeds closest to $v_{t,ref}$, with complementary sets defined by:

$$\begin{aligned} &\left| \hat{s} - \hat{s}_{o_{tl}} \right| < d_s \text{ or } \hat{v}_{t,r,l\pm1} \notin \left[v_{t,cl}, v_{t,ch} \right] \\ &\text{or } \left| \hat{v}_{t,r,l\pm1} - v_{t,ref} \right| = \min \left| \hat{v}_{t,r,i} - v_{t,ref} \right|, i = [1,...,N_l] \end{aligned} \tag{3.10}$$

then a forced lane change will be assigned. Here, d_s is a safe headway distance between the ACV and the preceding OV which will be defined in the next section (see equation (3.13)). A forced lane change is activated by adjusting the assigned speed for those lanes with following or leading maneuver outside the acceptable speed range. The adjustment is given by:

$$v_{t,r,j} = k_l \hat{v}^s_{t,o_j}, \hat{v}^s_{t,o_j} \notin \left[v_{t,cl}, v_{t,ch} \right], \; k_l \in \left[0, \frac{v_{t,\max}}{\hat{v}^s_{t,o_i}} \right] \tag{3.11}$$

where k_l is the adjustment factor and $v_{t,\max}$ is the maximum speed of the ACV. k_l can be selected to generate a high value of the objective function associated with tracking the adjusted reference speed of specific lanes. This will then force the MPC to track other lanes with closer assigned reference speed to the desired cruising speed $v_{t,ref}$ with lower values of the objective function. An example of the reference speed assignment and adjustment for multiple-lane scenario is shown in Figure 3-4.

In summary, the configurations of rule-based reference speed automatons for FSMs are listed as in Table I. In the rule description, the symbol " ∩" represents intersection, " ∪" means union, and the superscript "C" represents complement.

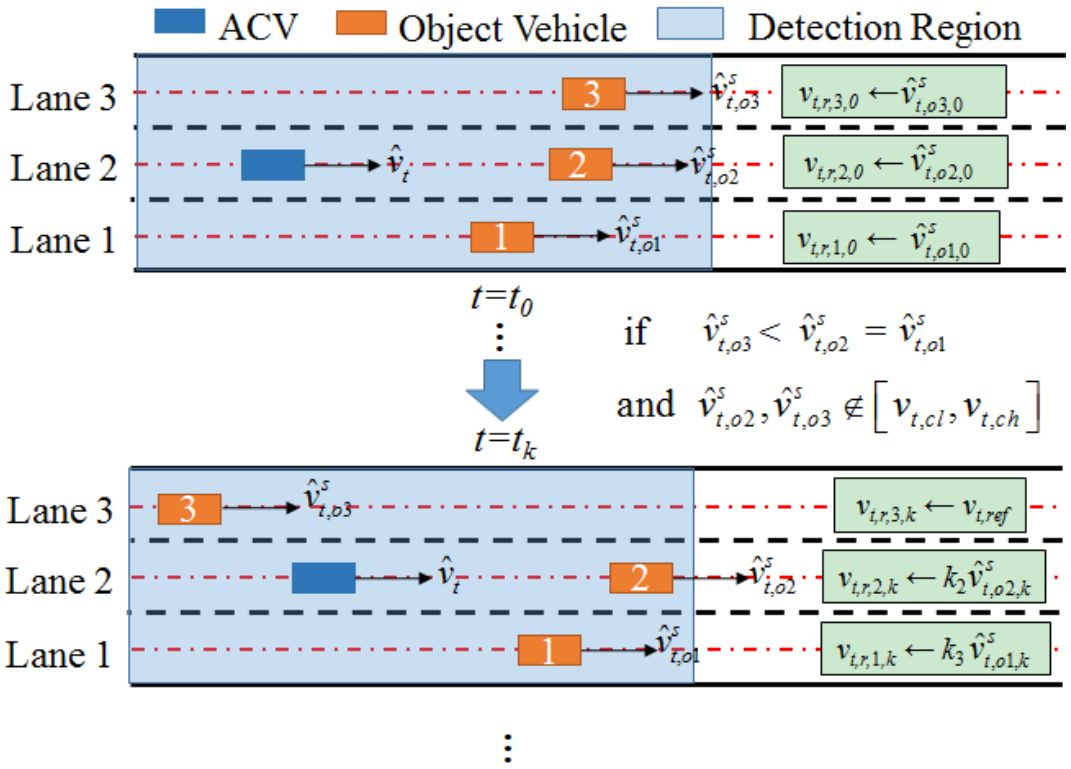


Figure 3-4: Prediction of reference speed assignment and adjustment

Table 3-1: Configuration of the Rule-Based Reference Speed Automaton for Lane *l*

Switch Set	**Rule Description**
$R_{l1,l2}$	(3.3)∩(3.4)∩(3.5)
$R_{l2,l1}$	$(3.3)^C$∪$(3.4)^C$ or (3.9)∪(3.10)
$R_{l1,l3}$	(3.3)∩(3.4)∩(3.5)
$R_{l3,l1}$	$(3.3)^C$∪$(3.4)^C$ or (3.9)∪(3.10)
R_{l_3,l_2}	(3.3)∩(3.4)∩(3.5)

3.3.3 MPC based Maneuver Planning

The MPC is configured in this section with the ability to conduct predictive lane change maneuver planning and guidance, with advance knowledge of the predicted reference speed assignment as described above.

The constraint to keep a safe distance between the ACV and any nearby OV *i* with the predicted uncertainties is tightened by the following elliptic inequality:

$$\left(\frac{y_e - \hat{y}_{e,o_i}}{\Delta y_{e,o_i} + f_{\sigma,y}\sigma_y}\right)^2 + \left(\frac{s - \hat{s}_{o_i}}{d_s}\right)^2 \geq 2 \tag{3.12}$$

$$d_s = \Delta s_{o,ss} + f_{\sigma,s}\sigma_s + f_{\zeta,Do}\zeta_{Do} \tag{3.13}$$

This constraint is depicted in Figure 3-5. A rectangular region is inscribed in the ellipse. Its dimensions $\Delta y_{e,oi}$ and $\Delta s_{o,ss}$ are calculated by incorporating the geometry (length and width) of OV *i* and the ACV. $\sigma_s = \sigma_{s,ACV} + \sigma_{s,oi}$ and $\sigma_y = \sigma_{y,ACV} + \sigma_{y,oi}$ are the combined

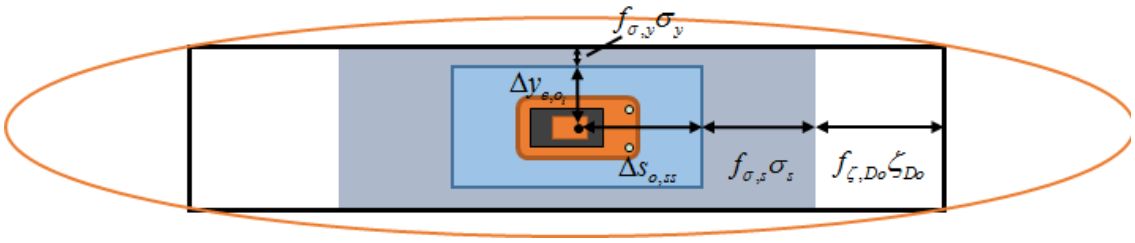


Figure 3-5: Elliptical boundary for collision avoidance with combined uncertainties σ_s, σ_y of ACV and OV

covariances of, respectively, the arc length and lateral position error of ACV and OV *i*, based on the predicted covariances from the motion prediction module. $f_{\sigma,s}$ *and* $f_{\sigma,y}$ define the cross-belief region of the combined states. For example, $f_{\sigma,s}$=$f_{\sigma,y}$=3 approximates a 99% cross-belief region between the arc length and lateral position. ζ_{Do} is a slack variable that allows the solver to find a feasible solution in emergency situations, e.g., full braking, see [31] for detailed discussions of this. Its reference value is v_t and we give it the auxiliary dynamics:

$$\dot{\zeta}_{Do} = u_{\zeta_{Do}} \tag{3.14}$$

In (3.13), $f_{\zeta,Do}$ is an tuning parameter (has a unit of time, typically $f_{\zeta,Do} > T_{mpc}$). To ensure a safety headway distance (exclude extreme events between MPC update intervals T_{mpc}) the following constraint should be satisfied:

$$\zeta_{Do} \geq T_{mpc} v_t \,/\, f_{\zeta,Do} \tag{3.15}$$

Other state constraints like friction ellipse of a real vehicle's tire/road contact, lane boundaries, speed limits and the minimum turning radius, etc. are also considered with uncertainties, for complete details, please refer to [31].

As uncertainties are considered, the lane selection maneuver planning problem to be solved over the prediction horizon [0, H_p] results in a stochastic MPC problem that is formulated by:

$$\min_{x_k,u_k} E\left(\sum_{k=1}^{N_p}\sum_{l=1}^{N_l}\left\|z_{l,k}\left(y_{1,k}-r_{1,l,k}\right)\right\|_{P_1}^2+\sum_{k=1}^{N_p}\left\|y_{2,k}-r_{2,k}\right\|_{P_2}^2+\sum_{k=0}^{N_p-1}\left\|u_k\right\|_R^2\right) \tag{3.16}$$

$$\text{subject to :} \qquad \dot{x}=f\left(x,u,w\right),\ x\in X,\ u\in U,\ w\in W \tag{3.17}$$

$$x(0)=x_0 \tag{3.18}$$

$$\sum_{l=1}^{N_l} z_l=1,\quad z_l\in\left[0,1\right],\quad \forall l\in\left\{1,...,N_l\right\} \tag{3.19}$$

$$\Pr\left(c_1\left(x,u\right)\geq 0\right)\geq\delta \tag{3.20}$$

$$c_2\left(x,u\right)\geq 0 \tag{3.21}$$

Here, the cost function minimizes the expectation of state tracking error and control efforts. x covers all the state variables of the ACV particle motion model given by (3.1), the slack variables in (3.14) and lane selection variables z_1~ z_{Nl} whose dynamics is described below in (3.23). X represents the state-space for x. x_0 denotes the current/initial state (measured and estimated). The estimation of the system outputs, namely the speed v_t and lateral position y_e of the ACV are grouped in vector y_1, and the slack variable outputs ζ_{Do} is in y_2. $r_{1,l}$, r_2, are, respectively, the candidate output references for lane l and references for the slack variables (corresponding respectively to y_1 and y_2). P_1, P_2 and R are the weighting matrices for the candidate maneuver tracking error, slack variable reference tracking error

and control efforts, respectively. In (3.17), U denotes the admissible set for input u, which includes the input to ACV motion model and selection variables. W is the state space for noise/disturbance w defined in (3.1). All the nonlinear inequality constraints with uncertain states such as the collision avoidance are included in the probabilistic constraint to satisfy a belief coefficient δ, denoted by (3.20). These constraints need to be tightened into deterministic constrain for calculating the solution, for example (3.12) is the tightened collision avoidance constraint and $f_{\sigma,s}$, $f_{\sigma,y}$ can be writen in a function of δ based on accumulated Gaussian distribution [78, 86]. Other deterministic inequality constraint like (3.15) are included in (3.21). The continuous model (3.17) is eventually discretized in sample steps ΔT, $\Delta T = H_p/N_p$ and H_p is the horizon length.

In order to realize the optimization-based lane selection, we utilize a suite of selection variables z_l in (3.19) to coordinate the consideration of tracking different lanes and their corresponding assigned reference speeds (by the pre-planning described in the previous section). The constraint (3.19) comes from an approximation method we proposed in [88] to relax the original MINP problem with the integrality constraint

$$\sum_{l=1}^{N_l} z_l = 1, \;\; z_l \in \{0,1\}, \;\; \forall l \in \{1,...,N_l\} \tag{3.22}$$

which assumes a binary lane selection of either 0 or 1. Here, with the relaxed formulation (3.22), the selection variables z_l are regarded as additional states with the auxiliary dynamics (included in (3.17)):

$$\begin{cases} \dot{z}_l = u_{z_l}, \text{ if } l \neq N_l \\ \dot{z}_l = -\sum_{i=1}^{N_l-1} u_{z_i}, \text{ if } l = N_l \end{cases} \tag{3.23}$$

Remark 3-2: To solve a MINP problem efficiently at real time, two fundamental approximation methods can be applied in tandem: relaxation and constraint enforcement [35]. The relaxation approach is to extend the feasible solution set of the problem, by relaxing or neglecting certain constraints, e.g., relaxing the integrality constraint from (3.22) to (3.19). Afterwards, constraint enforcement can be sought to exclude the solutions that are feasible under the relaxation but not for the original problem. For our problem formulation, the constraint enforcement is deemed optional as the relaxation of the integrality constraint will not affect the global minimum of (3.16). The optimization will naturally converge to tracking only one of the lanes if the configuration space is convex. If it is not, we have the following case.

Remark 3-3: With the relaxation of the lane selection variable involved in the integrality constraint, the ACV is no longer strictly guided to track only one of lanes. This may lead to undesired behaviors of the ACV in complex traffic scenarios with few available lanes, where the ACV may laterally approach an adjacent OV and stay in between two lanes until the OV is overtaken, as shown in our previous results [88]. More rules designed in the reference speed automaton for these situations may help to improve or exclude such behavior. The proposed rule-based speed assignment over the whole prediction horizon is meant to address this issue.

The computational complexity of our MPC formulations can be estimated. For our purposes, we solved the nonlinear programming problem above using the ACADO Toolkit [90] which implements qpOASES, a one-iteration SQP algorithm employing an active set strategy. From [44], the computational complexity of solving the MPC problem for tracking only one reference lane and reference speed with a prediction horizon length N_p is at most $O(g(N_x, N_u, N_p, N_c))$, where:

$$g\left(N_x, N_u, N_p, N_c\right) = {N_x}^3 + {N_u}^2 + {N_p}^2 + (N_u N_p)^2 + N_c N_u N_p \tag{3.24}$$

which is in low-order polynomial time. Here, N_x is the number of states, N_u is the number of control inputs, N_p is the length of the prediction horizon, N_c is the number of constraints. The computational complexity of solving the MINP problem is then $(N_l^{Np} g(N_x, N_u, N_p, N_c))$, which is in high-order polynomial time, since N_p is typically in the order of 40 or more for the present application. The computational complexity of the resulting NLP problem with the approximation method is at most $O(g(N_x+N_l, N_{u+}N_l\text{-}1, N_p, N_c+N_l+1))$, which is in low-order polynomial time. Therefore, by adopting the approximation method, the complexity of solving the MINP problem can be significantly reduced to an NLP problem with much less computational burden. As we comment in the next section, the resulting execution times are feasible for real-time implementation in the many scenarios we have tested.

3.4 Simulation Results and Conclusion

3.4.1 Simulation Results

To illustrate the performance of the predictive maneuver planning approach, we consider a straight six-lane highway where the ACV faces sequentially connected scenarios,

like overtaking, following and collision avoidance in the presence of eight nearby object vehicles (OVs), as shown in Figure 3-6. The situations progress from those requiring simple responses (lane change) to aggressive ones that push the vehicle dynamics and the control to the limit. The proposed approach, hereafter labeled OSM (for optimized sequence of maneuvers within a prediction horizon), will be compared with the previous approach in [88] where only one optimized maneuver (labled OOM) is selected for the entire prediction horizon. We also compare the results to a common adaptive cruise control (ACC) scheme to show the advantage of two-dimensional maneuver planning. The assumed uncertainties for the ACV and OVs motion models are given in Table 3-2, where the disturbances/noises are modelled by normal distribution $N(\mu,\sigma^2)$ with mean μ and covariance σ^2. The parameters selected for the MPC formulation are listed in Table 3-3.

Table 3-2: Uncertainties and Parameters for Estimation of ACV and OV

Vehicle	**Parameter**	**Value**	**Vehicle**	**Parameter**	**Value**
ACV	w_{a_t} [m/s^2]	$N(0,0.01)$	**OV (1~8)**	$w_{\dot{s},o}$ [m/s]	$N(0,25)$
	$w_{\Delta\dot{\psi}_d}$ [rad/s^2]	$N(0,10^{-4})$		$w_{y,o}$ [m]	$N(0,1)$
	v_s [m]	$N(0,0.01)$		$v_{s,o}$ [m]	$N(0,25)$
	v_{y_e} [m]	$N(0,0.01)$		$v_{y,o}$ [m]	$N(0,1)$
	v_{a_t} [m/s^2]	$N(0,0.01)$		K_{s1}	2.5
	$v_{\dot{\psi}_p}$ [rad/s^2]	$N(0, 10^{-4})$		K_{s2}	2
				K_{y1}	2.5
				K_{y2}	2

Table 3-3: Main Parameters Selected

Parameter	Value		Parameter	Value		Parameter	Value
$v_{t,ref}$ [m/s]	30		$\Delta s_{o,ss}$ [m]	5.3		$P_{2,\zeta_{gg}}$	20
Δv_t [m/s]	2.5		$\Delta y_{e,o_i}$ [m]	2.3		$P_{2,\zeta_{do}}$	20
Δy_e [m]	1.85		g [m/s^2]	9.8		R_{a_t}	50
T_{at} [s]	0.075		N_p	40		$R_{\Delta\dot{\psi}_p}$	250
$T_{\psi p}$ [s]	0.2		ΔT [s]	0.15		$R_{u_{\zeta do}}$	0.001
T_d	7s		k_v	0.8		$R_{u_{\zeta gg}}$	0.001
$f_{\zeta,Do}$	0.5		$P_{1,ye}$	3		$R_{u_{zl}}$	100
$f_{\sigma,s}$, $f_{\sigma,y}$	3		$P_{1,vt}$	2			

The longitudinal speeds of the OVs are set to be constant at values to be discussed below for the various scenarios. The measurement sampling time/MPC update time T_{mpc} is set to 150ms. The state trajectories for the whole scenario are shown in Figure 3-6 and Figure 3-7.

3.4.1.1 Preceding OVs detected at Cruising (In the first 20 s)

In this scenario, the ACV initially occupies lane 1 (right most lane) at its reference/desired speed of 30 m/s. Then, it faces two slower vehicles OV1 and OV2, going in parallel in lane 1 and lane 2 at speed of 25 m/s. At around time of 7 seconds, the ACV under OSM starts a lane change to lane 3, which is the closest available lane, to overtake

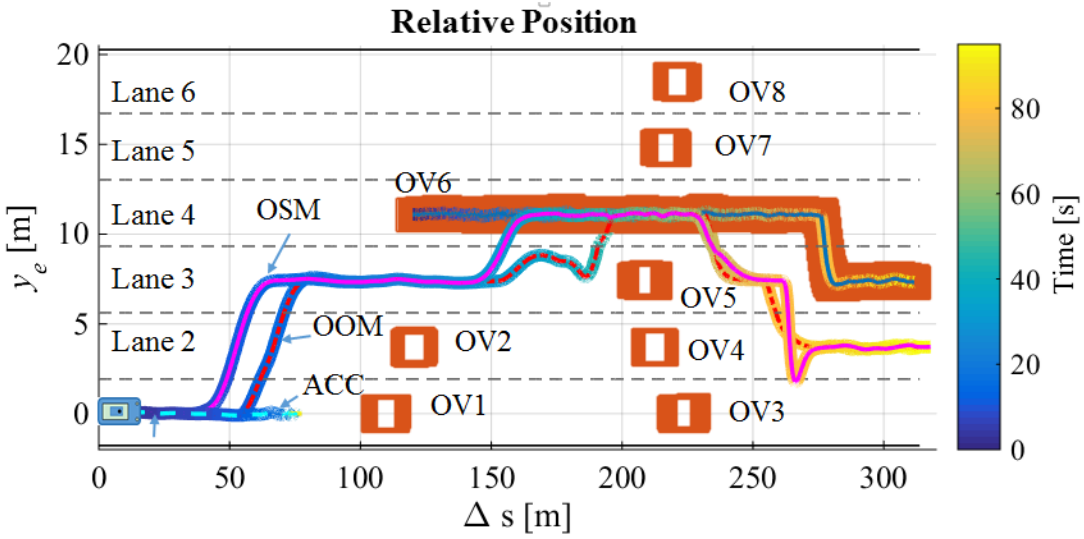


Figure 3-6: Relative position of ACV and OVs in the simulation

OV1 and OV3. However, for the case with OOM, the lane change to lane 3 happens around 10 seconds. But the ACV under ACC will only slow down starting from 3 second to follow OV1 as lane change is not available under this setting. This also shows the potential undesired local minimum for maneuver planning. The predictive speed assignment in the OSM case considers reference speed change on lane 1 as well as the potential reference speed change on lane 2. This can be also seen in Figure 3-8 which shows the computed trajectories for the prediction horizon at t=11.1s, including the planned relative position, the reference speed assignment and the lane selection variables. We can see when ACV approaches OV2 (at 11+2= 13s), the reference speeds of the OSM case in the prediction horizon (set via the rule-based reference speed automaton) for lane 2 vary from the desired cruise speed to those of speed of the OV2. This is done as soon as the gaps between the ACV and these OVs are predicted to be smaller than the threshold defined in equation

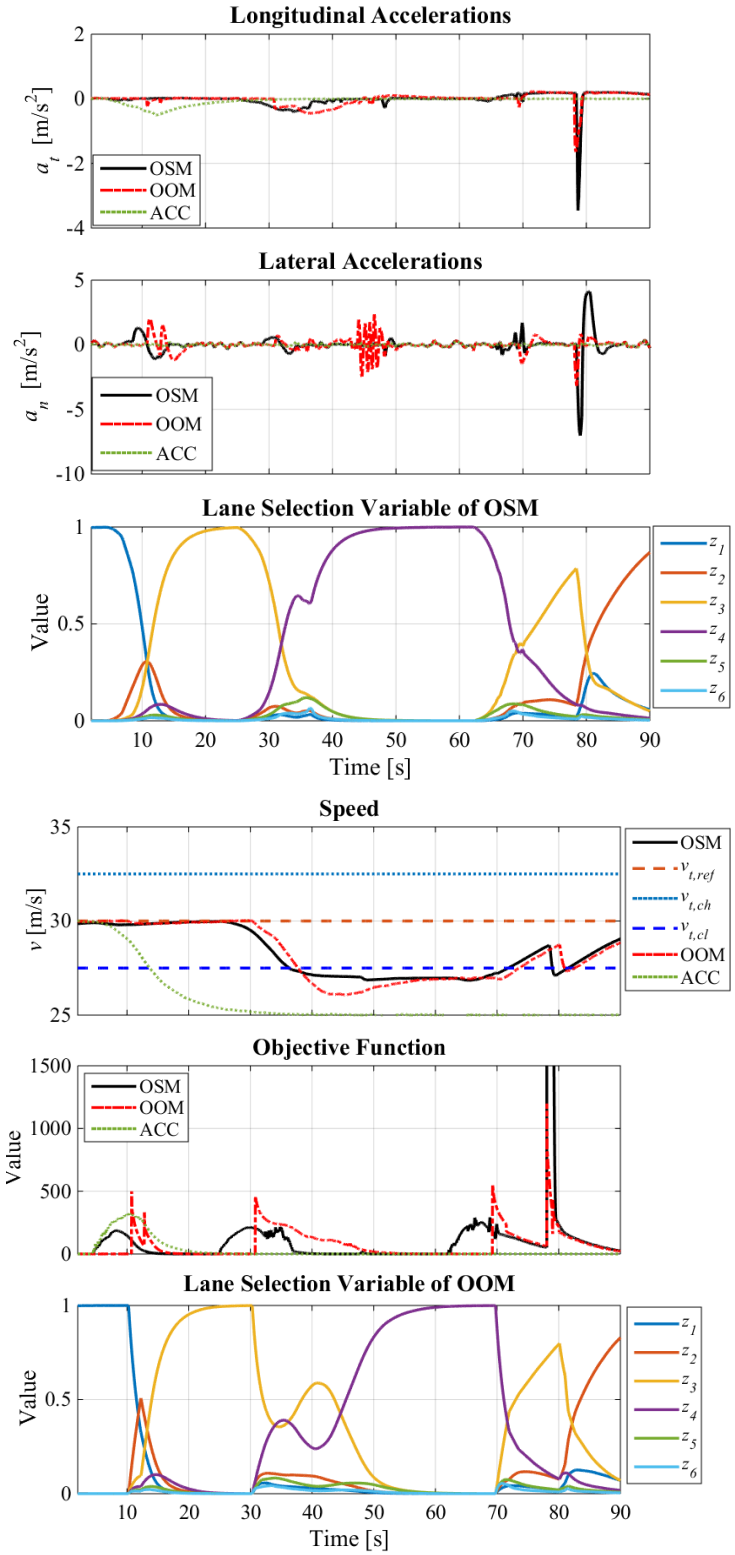


Figure 3-7: State evolutions for the whole duration under the different manuever

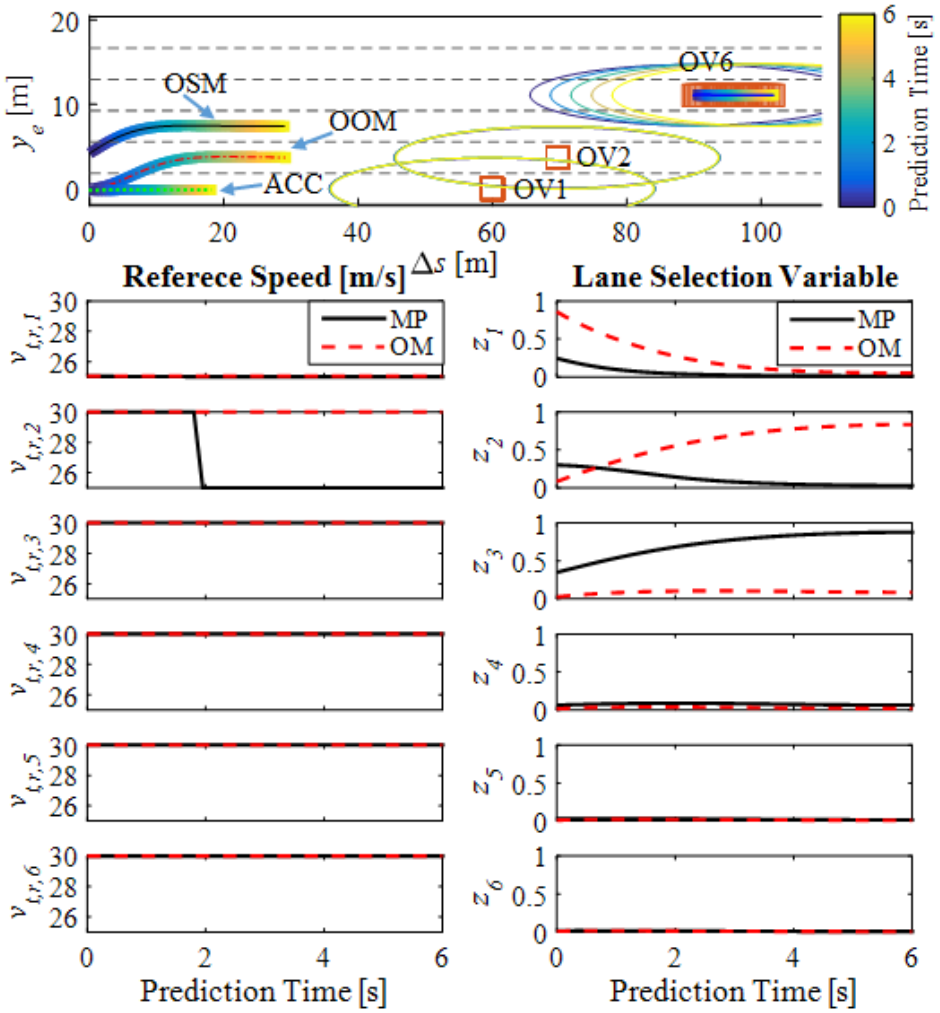


Figure 3-8: Trajectory examples for the predictive horizon at t=11.1s

(3.10). For the same horizon, the OOM case switched only the reference speed for lane 1. The changes of the speed references in the prediction horizon are captured via the switching of $v_{t,r,1}$, $v_{t,r,2}$ and $v_{t,r,2}$ at the left column. These changes gradually increase the value of the objective function and force the ACV to either slow down to follow OV2 or change lane. The weighting parameters listed in Table III promote a lane change maneuver if an open lane is available. The ACV is predictively controlled to steer to lane 3 in this case. The pre-

planning of the reference speed changes helps to consider changes in the environment from the beginning and helps the MPC to generate more smooth trajectories.

Note that during the lane change, the rise of z_2 is observed in Figure 3-7, this is due to the fact that ACV needs to go across lane 2 to reach lane 3 and the rise of z_2 actually reduces the value of the objective function. However, z_2 will not rise to 1 because settling down in lane 2 is not the local minimum for that moment. The MPC update continues to predictively change lane to lane 3 further reducing the objective function to zero.

3.4.1.2 Overtaking While Following

For the next 40s (20~60s), the ACV faces a "traffic jam" consisting of OV3 in lane 1, OV4 in lane 2, OV5 in lane 3, OV6 in lane 4, OV7 in lane 5 and OV8 in lane 6. Only lane 4 is eventually available to go through. However, the ACV has to change lane to follow a slower OV6 first, because it occupies lane 4 in front of the ACV. As OV6 is faster than the other OVs, it can pass through the "block" together with the ACV. Afterwards, the ACV will switch from a following mode (discrete state) to lane change mode in order to overtake the OV6. The details of this operation are as shown in Figure 3-6, Figure 3-7 and Figure 3-9.

From Figure 3-9 , we can see the detailed workings of the ACV detects the "traffic jam" around time=30.9s. Similar as the previous scenario, the ACV with OSM can predict the reference speed switch. The reference speed of the lanes from lane 1 to lane 6, except for lane 4 (which tracks a higher speed 27m/s of OV6), in the predictive horizon switch from 30m/s to 25m/s at different times when the ACV approaches the jam. While in the OOM case only the thresholds of lane 2, 3 and 4 are triggered based on the relative position

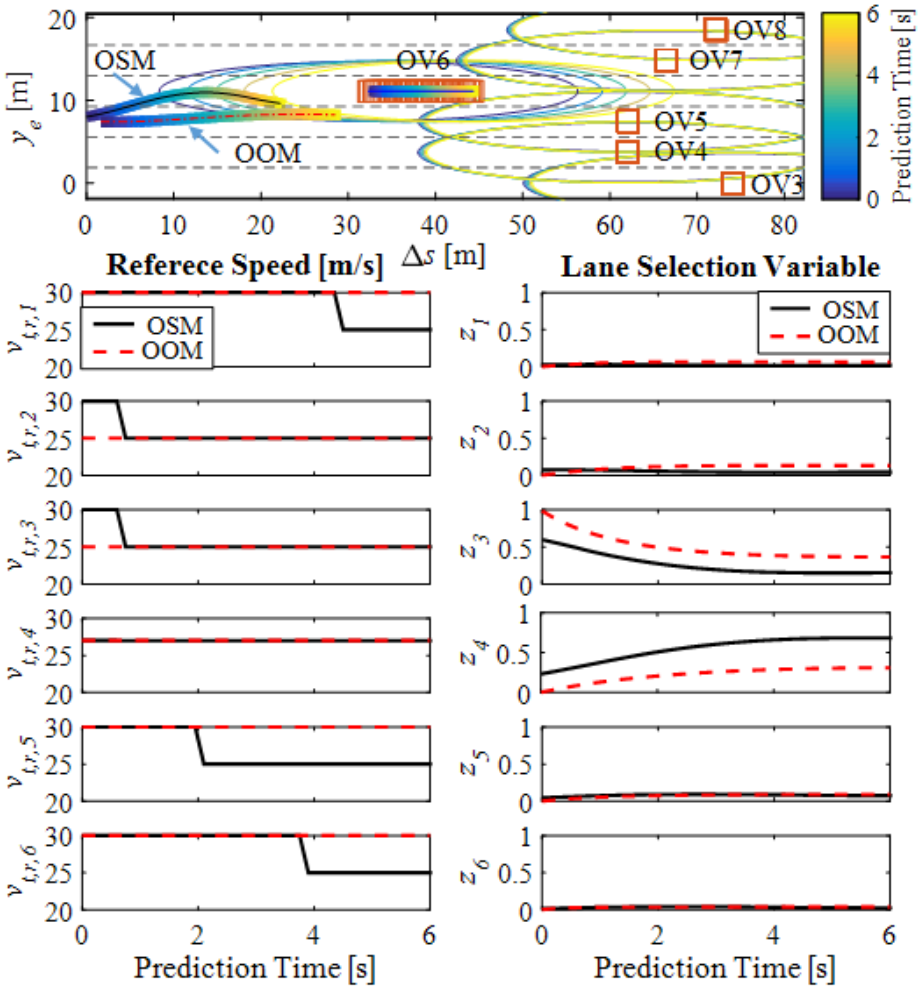


Figure 3-9: Trajectory examples for the predictive horizon at t=30.9s

at that moment. The different speed assignments in the two cases affect their position planning, as shown in the top of Figure 3-9. For case of OSM, the ACV plan to change lane to lane 4 and follow OV6 with least speed gap to minimize the objective function (zero value local minimum can be achieved). However, in the OOM case, the ACV only has a sideward movement to OV6 to reduce the object function but not strictly following one lane (see the z_3 and z_4 in Figure 3-9). This lead to the dilemma of the ACV at the joint area of the elliptical boundaries of OV6 and OV5 (non-zero local minimum), which further

causes the oscillation of lateral acceleration, lane selection, more speed reduction and slow settling of the objective function between 40s and 50s in Figure 3-9. Here we can see the predictive maneuver sequence can help achieve a feasible and smoother motion plan which manifests as reduced occupant discomfort and mechanical wear. Finally, when OV6 passes OV5, the ACV gradually goes back to lane 4. Note that as the elliptical collision boundary with 99% uncertainty belief region is not violated during the planning, a collision free trajectory is achieved.

Afterwards, in both cases, the ACV follows the front OV6 until it passes the block/jam. When ACV exceed the other OV(3,4,5,7,8), reference speeds of the related lanes are then reduced by the coefficient of k_l to ensure that tracking these lanes leads to more cost. Therefore, the MPC will command another lane change to lane 3 to overtake OV6 and increase the ACVs speed to 30m/s. Finally, the reference speed of each lane will switch back to the desired cruise speed when ACV passes all the OVs.

3.4.1.3 Collision Avoidance

In the last 15 s, OV6 makes an unexpected sudden lane change starting at time=78 s to the lane occupied by the ACV when the ACV is just overtaking the OV6 from behind. To avoid collision with OV6, the ACV needs to plan its trajectories without entering the elliptical boundary, which might require either slowing down (or speeding up in other situations) in the longitudinal direction and lane change in the lateral direction. The spike in the objective function is mainly due to the sudden braking by the ACV.

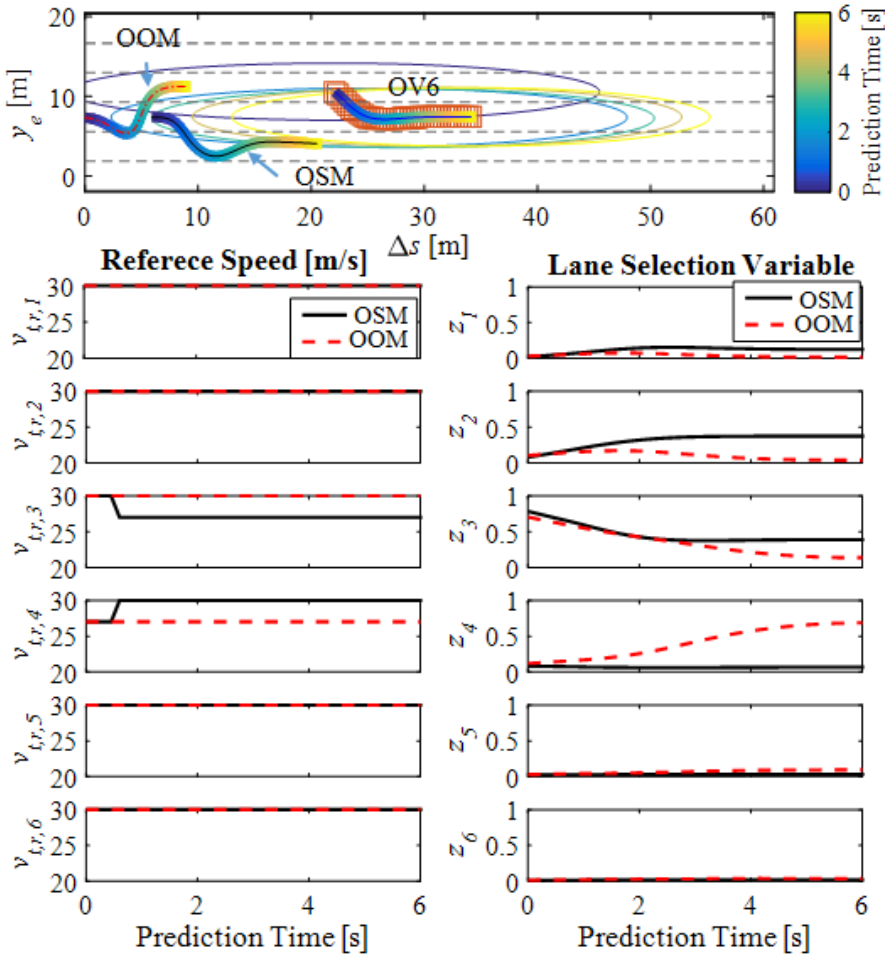


Figure 3-10: Trajectory examples for the predictive horizon at t=78.45s

From Figure 3-7, the ACV under OSM and OOM combines both decelerating and lane change to the right to avoid the collision. The future trajectory of the OV6 is predicted at each sampling time, an example for the prediction horizon at time=78.45 is shown in Figure 3-10. Based on this, in OSM case, the lane reference speeds from lane 4 to lane 1 are assigned with OV6's speed at the time OV6 is predicted to go across them. The ACV in this case plans to change lane to lane 2 to avoid a collision with lowest cost. While the OOM case can only assign the reference speed based on the estimation of the current lane occupation for OV6, the speed assignment can't match the position prediction of the OV6,

therefore its planning decision is more naïve than OSM case. In this case, slowing down and changing lane to lane 4 to track the speed of OV6 is with the lowest cost. When OV6 goes across the boundary between lane 3 and lane 4, change lane to lane 2 became the best planning decision, which matches the results in Figure 3-6. In both cases, the collision probability during the collision avoidance are close to 0% due to the constraint (3.25) considering the uncertainties.

Here, we have to point out that the relative positions between OV6 and the ACV when OV6 starts to change lane will affect the maneuver planning results. A closer distance requires faster lane change. That's why the lane change of the ACV in OSM case is more aggressive than the ACV in OOM case. With predictive speed assignment, the ACV can be better prepared for the change of the environment and then make more appropriate planning decisions. If OV6 and ACV are too close to each other, there might be no feasible solution in the MPC that avoids entering the collision boundary.

3.4.1.4 Execution Time

Finally, we comment on the execution times involved in the above simulations. The MPC solver in the ACADO Toolkit is executed on an Intel Dual Core i5-4200M 2.4 GHz processor and 4GB RAM. The execution times for the MPC problem for this simulation are shown in Figure 3-11. Note that for all the cases compared, the MPC execution times are mostly in the order of 90ms or less; increasing when the elliptical inequality constraints are engaged, more sharply in the initial part of collision avoidance.

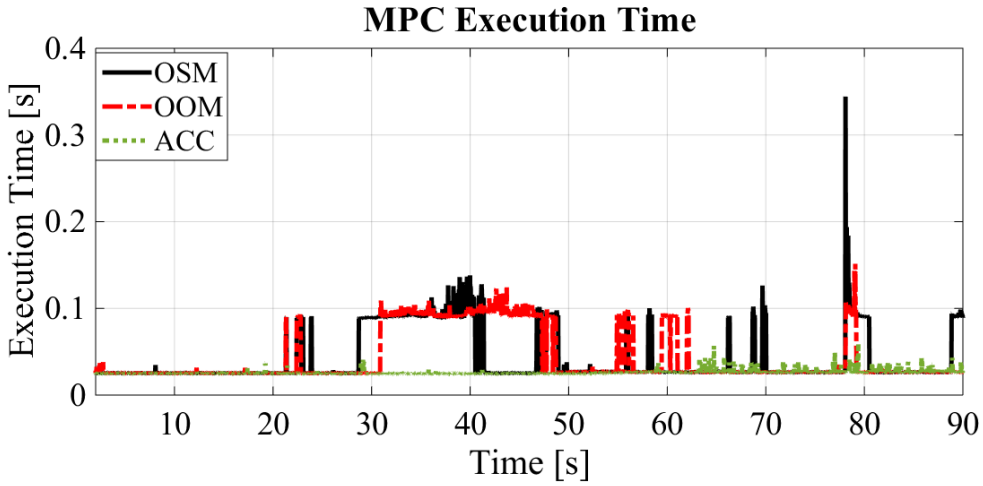


Figure 3-11:. History of MPC execution time

3.4.2 Conclusions

This chapter outlined, a predictive maneuver planning and control framework that integrates both discrete maneuver planning and motion trajectory planning for an autonomous controlled vehicle in the presence of uncertainties (disturbances and sensor noise). Within the prediction horizon, a rule-based assignment of reference speeds for each reference lane is applied in each interval of the horizon based on the predicted motion of the autonomous vehicle and other object vehicles. Then, the sequence of maneuvers is incorporated in a relaxed stochastic MPC formulated to simultaneously generate the optimized reference selections and control input trajectories that minimize an objective function subjected to traffic constraints and rules involving other objects that are prevalent in public traffic. A series of simulation experiments showed that, the maneuver planning helps the autonomous vehicle to better accommodate the environment. Also, the modification in the reference speed assignment improves the optimality and the robustness

of the maneuver decision in trajectory planning without adding computational complexity to the optimization problem.

CHAPTER 4

OBSTACLE FILTERING AND VEHICLE GROUPING ALGORITHM FOR AUTONOMOUS DRIVING

4.1 Abstract

This chapter presents an obstacle filtering algorithm and vehicle grouping algorithm that mimics human driver-like grouping of objects within a model predictive control scheme for an autonomous road vehicle. In the algorithm, a time to collision criteria is first used as risk assessment indicator to filter the potentially dangerous obstacle object vehicles in the proximity of the autonomously controlled vehicle. Then, the filtered object vehicles with overlapping elliptical collision areas put into groups. A hyper elliptical boundary is regenerated to define an extended collision area for the group. This grouping serves to compute the time varying areas that are to be occupied by vehicle groups in the predicted motion plan so that those areas including the undesired local minimums can be excluded for the non-convex motion planning problem. To minimize conservatism, the parameters for the tightest hyper ellipse are determined by solving an optimization problem. To reduce the computational burden of the grouping and boundary generation for online implementation, supervised learning methods are applied to train the neural networks that compute the optimal boundary of the group (s) that maximizes the available planning field. The proposed algorithm is incorporated in a predictive guidance scheme and its performance and computational details are illustrated via simulations of an autonomously

controlled vehicle in public highway traffic scenarios involving multiple other object vehicles.

4.2 Introduction

In the march towards autonomous driving, the task of guiding the controlled vehicle to rapidly and systematically accommodate the plethora of changing constraints in public traffic, from tire/road friction conditions to avoiding stationary and moving obstacles, while obeying the traffic rules, signals and so on, remains a major challenge. Recently, there are two main research directions being pursued to address the control problem of autonomous driving: learning based control and planning based control. Learning based control uses supervised learning methods to train a complex policy function such as a convolutional neural network (CNN), based on the paired data records of input s(sensing data [4, 5, 6] /machine vision [10, 9]) and outputs (driver's control signals) of the test vehicle, and then apply the trained CNN on the autonomously controlled vehicle (ACV) or ego vehicle. However, performance of this approach suffers from the large size of the training data needed to represent a variety of driving scenarios in order to approach 'guaranteed' safety.

The planning based control replaces the NN with a hierarchical control scheme combining a motion planner and a controller. The planner uses the perception information to plan the motion of the ACV while the controller controls the vehicle to follow the plan. In this category, model predictive control (MPC) is receiving significant attention, not only because the problem formulation can naturally accommodate the changing constraints in public traffic, but also because its finite receding horizon optimization scheme models

human drivers very well [57]. In [65], MPC is applied for static obstacle avoidance and works in [29] formulates it as a local reactive controller for trajectory planning to simultaneously track the path and avoid dynamic obstacles. In [43] [31], the motion planning and guidance of ACVs are formulated for general public traffic scenarios by adopting lane centerlines as reference paths and uniformly expressing the motion of the controlled vehicle and all other objects, traffic rules/signs, lane limits, and road friction limits within the prediction horizon. The framework is extended later by this chapter to the case of multi-lane scenarios by first structuring the controlled vehicle's maneuvers in finite state machines which lead to a hybrid system framework, where rule-based [89] and optimal maneuver selections [88] are then sought.

The performance of MPC applied to autonomous driving relies on the constraint descriptions involving the motion of the ACV as well as of the other object vehicles (obstacles). Under some practical assumptions and an appropriate physics-based model of the motion of object vehicles and the information from perception, the expected evolution of the motion of the object vehicles can be predicted for a certain future horizon [65]. For instance, physics-based motion models assuming constant velocity/acceleration can be used to predict the motion of object vehicles for a short future horizon (order of 1 second). Moreover, by taking advantage of frequent update of the MPC (5~10Hz), such models can also be used for longer term predictions (4~6s) [31], provided computations can be completed sooner. Other proposed models includes maneuver-based [69, 68] and interaction-aware [70, 71] motion prediction models under the consideration of the driver's maneuver intention and the inter-dependencies between the individual vehicles' maneuvers.

These latter models are complex and require lots of real time test data for model identification.

Group tracking is a technique used in multi-object tracking applications (e.g. people tracking) that attempts to simplify the data association problem when the position measurements for individual objects are very close together [91]. The group formation process can reflect the interactions among the objects. Multi-hypotheses tracking [92] is one method that can be used to track groups by making hypotheses of group behaviors like merging or splitting. In our previous work [93], we proposed a grouping concept to describe the collision avoidance constraints of object vehicles that are close to each other by proposing an optimization-based group boundary generation. The work showed that for the case with crowded object vehicles the group description with proper boundary design can help to reduce the efforts of evaluating the constraint avoidance for the MPC problem and also to exclude the undesired local minimums at the intersections of the collision boundaries of individual object vehicles. The approach also effectively excludes undesired maneuver selections for the hybrid predictive planning problem (with both maneuver and motion planning). However, in [20] , the group formation model was not introduced rigorously. Furthermore, the computational complexity of the optimization used to generate the group boundary turned out to be so significant that it can reduce the applicable update rate of the MPC and thereby worsen the performance of predictive control with physics-based motion prediction models.

When formulating obstacle avoidance constraints for the prediction horizon, it is possible to model the dynamic motion of surrounding obstacle vehicles. However, to do

this, one invariably needs to impose some assumptions about the unknown future inputs to these obstacle vehicles, inputs which are not generally available to the ACV controlled by MPC. However, by using the latest information about obstacles and the environment constructed from available sensing via radar, lidar, camera and V2V or V2I communications, one can minimize the required complexity of the models needed to describe the motions of the object vehicles. This can in turn help to reduce the computations of the MPC so that they can be completed fast enough and then take advantage of frequent updates. Using the MPC internal time as a state variable and the latest accelerations, speeds and positions of obstacle objects obtained by sensing or communication, one can derive algebraic descriptions of the motion of the geometries representing obstacle object vehicles for the whole prediction horizon [31].

There are several ways of modeling the geometric descriptions in the 2D configuration space [49], including e.g. polygonal models [94], described by the combination of linear curves; semi-algebraic models, like polynomials; or algebraic models like circles, ellipses [95] or hyper-ellipses [96]. Algebraic models are more efficient in describing obstacles with multi-edges since they generally need fewer parameters to be specified. For example, for describing a rectangular obstacle (4 edges), applying linear curves requires 8 parameters, while only 4 parameters are required for a conservative ellipse or hyper-ellipse. In our previous work [43] [89] [88], ellipses are used to describe the geometry of static/dynamic vehicular obstacles for MPC-based motion planning. It can be argued that ellipses naturally and conservatively describe the 2D geometry of modern road vehicles. However, possible overlaps in the prevailing distribution of the

obstacles/ellipses may create undesirable local minima (or global minima for the finite horizon planning problem), which may trap the ACV. In addition, in the presence of more obstacle object vehicles around the ACV, the total number of evaluations for constraint violation/collision detection increases, which increases the complexity of, and the execution times needed for solving the optimization problem at each MPC update.

In this chapter, we propose a concept of obstacle filtering concept and algorithm for the prediction of the motion of obstacle vehicle objects around an autonomous vehicle in public traffic. The algorithm may mimic human driver like cognitive actions [97] and covers three procedures: risk assessment, obstacle grouping and group boundary re-generation. This algorithm adaptively refines the constrain set to create a configuration space that excludes undesired local or global minima from possible overlaps of elliptic geometries, thus improving the performance of the MPC optimization solver in finding the best solution for the motion plans. Then, based on the group classification method and optimization based group boundary generation method, a supervised learning method is implemented to train a neural network that can provide the states of the vehicle groups, efficiently online, so that they can be used in the MPC for motion guidance. The estimates of the training requirements for the approach are also given and illustrate that this approach can avoid the computational limitations mentioned above, while retaining the core group modeling scheme.

4.3 Algorithm Design

4.3.1 Risk Assesment

Based on the motion prediction of the detected OVs, The risk here are associated with physical collision between the ACV and OVs, which is represented by the ACV entering the collision area defined around the OVs. Based on the kinematics model used to predict the motion of OVs, we use time to collision (TTC) T_c as an indicator to assess the risk of collision with in the detection range s_d, of the deployed sensors. Thus, we can define a range between ACV and OV i where a collision might happen along the reference path within a specified positive time T_c as:

$$\left| s_A - s_{o_i} \right| < s_d \tag{4.1}$$

$$0 < \frac{s_{o_i} - s_A}{v^s_{t,A} - v^s_{t,o_i}} < T_c \tag{4.2}$$

where s_A, $v^s_{t,A}$ are the longitudinal position and velocity of the ACV in the road frame. The OVs with their states satisfying both equation (4.1) and (4.2) will be considered to enough proximity to have potential danger of collision with the ACV, regardless of which lane they occupy.

4.3.2 Obstacle Grouping

In obstacle grouping, two step are followed. First, we need to determine if two OVs have intersecting collision areas. The sufficient condition for no overlapping of two ellipses with their axes (either major axes or minor axes) parallel to each other can be easily derived.

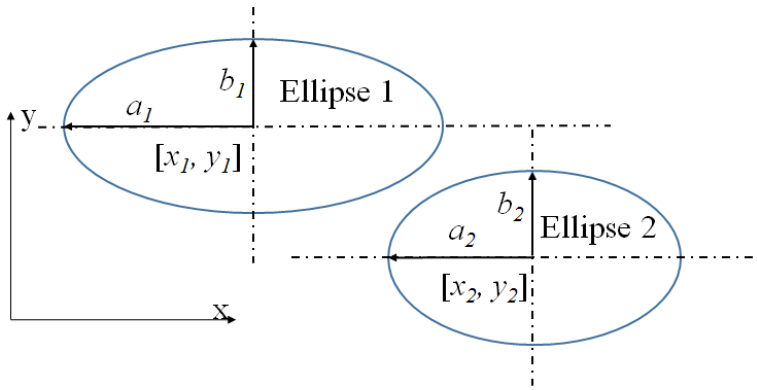


Figure 4-1: Definition of two ellipses with their axes parallel to each other

Second, this condition is applied to all the OVs filtered by the risk assessment step, to identify the groups and OVs belonging to each group.

Any two ellipses with their axes parallel to each other, as shown in Figure 4-1, can be defined by the following standard forms:

$$\left(\frac{x_{E1}-x_1}{a_1}\right)^2+\left(\frac{y_{E1}-y_1}{b_1}\right)^2=1 \tag{4.3}$$

$$\left(\frac{x_{E2}-x_2}{a_2}\right)^2+\left(\frac{y_{E2}-y_2}{b_2}\right)^2=1 \tag{4.4}$$

where $[x_{E1}, y_{E1}]$, $[x_{E2}, y_{E2}]$ are the points on the two ellipses. $[x_1, y_1]$, $[x_2, y_2]$ are the center of the two ellipses. a_1, a_2 are the half major axes of the two ellipses. b_1, b_2 represent the half minor axes of the two ellipses.

Starting with external tangentiality condition, it can be shown that the sufficient condition for two given ellipses to not overlap with each other is to simultaneously satisfy (4.5) and (4.6). See Appendix A for the derivation of these conditions.

$$\left(\frac{x_2 - x_1}{a_1 + \frac{a_2^{\,2} b_1}{\min[a_1 b_2, a_2 b_1]}}\right)^2 + \left(\frac{y_2 - y_1}{b_1 + \frac{a_1 b_2^{\,2}}{\min[a_1 b_2, a_2 b_1]}}\right)^2 \geq 1 \tag{4.5}$$

$$\left(\frac{x_1 - x_2}{a_2 + \frac{a_1^{\,2} b_2}{\min[a_1 b_2, a_2 b_1]}}\right)^2 + \left(\frac{y_1 - y_2}{b_2 + \frac{a_2 b_1^{\,2}}{\min[a_1 b_2, a_2 b_1]}}\right)^2 \geq 1 \tag{4.6}$$

Therefore, we can design a function J_o in (4.7) to identify the overlap condition of any two OVs *i* and *j* by comparing Jo with 2: if Jo≥2, the collision area of OV *i* and OV *j* don't overlap; if Jo<2, the collision area of OV *i* and OV *j* overlap.

$$\begin{aligned} J_o = &\left(\frac{s_{o,i} - s_{o,j}}{\Delta s_{o_j} + \frac{\Delta s_{o_i}^2 \Delta y_{e,o_j}}{\min[\Delta s_{o_i} \Delta y_{e,o_j}, \Delta s_{o_j} \Delta y_{e,o_i}]}}\right)^2 + \left(\frac{y_{e,i} - y_{e,j}}{\Delta y_{e,o_j} + \frac{\Delta s_{o_j} \Delta y_{e,o_i}^2}{\min[\Delta s_{o_i} \Delta y_{e,o_j}, \Delta s_{o_j} \Delta y_{e,o_i}]}}\right)^2 \\ &+ \left(\frac{s_{o,j} - s_{o,i}}{\Delta s_{o_i} + \frac{\Delta s_{o_j}^2 \Delta y_{e,o_i}}{\min[\Delta s_{o_i} \Delta y_{e,o_j}, \Delta s_{o_j} \Delta y_{e,o_i}]}}\right)^2 + \left(\frac{y_{e,j} - y_{e,i}}{\Delta y_{e,o_i} + \frac{\Delta s_{o_i} \Delta y_{e,o_j}^2}{\min[\Delta s_{o_i} \Delta y_{e,o_j}, \Delta s_{o_j} \Delta y_{e,o_i}]}}\right)^2 \end{aligned} \tag{4.7}$$

Then, we can define the group by the following statement: A group consists of set of OVs where for anyone OV in the group, there is another OV with a collision area overlapping with it.

4.3.3 Group Boundary Regeneration

After identifying the OV groups, a new collision area can be regenerated for the group to cover the collision areas of all OVs in the group and systematically exclude the

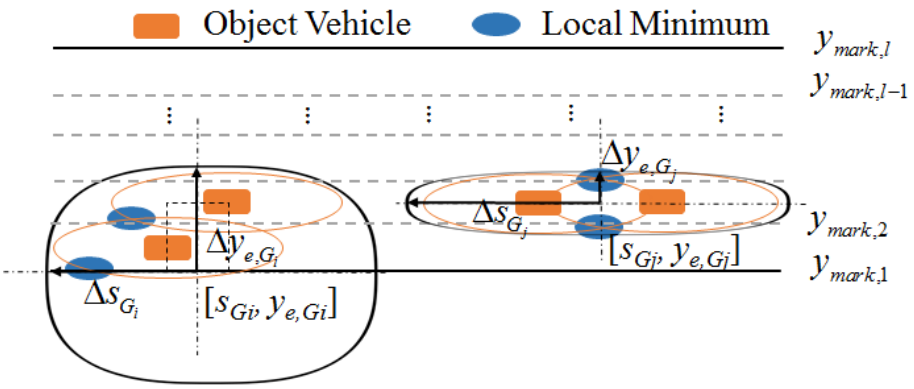


Figure 4-2: Example of 4th order hyper elliptical group boundary regeneration

undesired local and global minimums that come from overlapping elliptical intersections (Figure 4-2). Here, we use the 4th order hyper ellipse to re-generate the boundary. This algebraic geometry requires few parameters to characterize and define a continuous boundary for the conservative collision area of the group. Below, we shall seek the tightest description of this boundary that doesn't waste too much collision free space.

To being with, the 4th order hyper elliptical boundary for the ACV to avoid group *i* is defined as:

$$\left(\frac{y_{e,A}-y_{e,G_i}}{\Delta y_{e,G_i}}\right)^4+\left(\frac{s_A-s_{G_i}}{\Delta s_{G_i}}\right)^4\geq 1 \tag{4.8}$$

where s_{G_i},y_{e,G_i} are the center position of the group *i*, which can be obtained by taking the average of the longitudinal and lateral positions of the constituent OVs in the group. However, the lateral position y_{e,G_i}, also depends on the positions of the element OVs. If one of the OVs is on the side lane next to the road boundary, y_{e,G_i} can be placed on the road boundary to guide the ACV to the available lanes on the other side of the road and to

avoid creating local minimums at the intersections of the hyper elliptical boundary and the road boundary, as show in Figure 4-2 (left).

$$s_{G_i} = \frac{1}{N_{G_i}} \sum_{n=1}^{N_{G_i}} s_{o_n} \tag{4.9}$$

$$y_{e,G_i} = \begin{cases} y_{mark,l}, if \quad \exists y_{e,o_n} \in \left[y_{mark,l}, y_{mark,2} \right], n \in \left\{ 1,2,...,N_{G_i} \right\} \\ y_{mark,l}, if \quad \exists y_{e,o_n} \in \left[y_{mark,l\text{-}1}, y_{mark,l} \right], n \in \left\{ 1,2,...,N_{G_i} \right\} \\ \frac{1}{n_o} \sum_{i=1}^{n_o} y_{c,ei}, \ else \end{cases} \tag{4.10}$$

where N_{Gi} is the number of OVs in group i.

The half minor and half major axes $\Delta y_{e,G_i}$ and Δs_{G_i} of the tightest boundary of the group can be determined by posing an optimization problem. That is, we seek to find the hyper ellipse with minimum area that covers all the collision areas of the constituent OVs. As the area of a hyper ellipse is proportional to the product of the length of the major and minor axes, the optimization problem can be defined as:

$$\min_{\Delta s_{G_i} \Delta y_{e,o_i}} \Delta s_{G_i} \Delta y_{e,o_i} \tag{4.11}$$

subject to:

$$\begin{cases} \left(\frac{\mathbf{s}_{\mathbf{s},\mathbf{G_i}} - s_{o_1}}{\Delta s_{o_1}} \right)^2 + \left(\frac{\mathbf{y}_{\mathbf{e},\mathbf{s},\mathbf{G_i}} - y_{e,o_1}}{\Delta y_{e,o_i}} \right)^2 \geq 1 \\ \vdots \\ \left(\frac{\mathbf{s}_{\mathbf{s},\mathbf{G_i}} - s_{o_1}}{\Delta s_{o_{N_{Gi}}}} \right)^2 + \left(\frac{\mathbf{y}_{\mathbf{e},\mathbf{s},\mathbf{G_i}} - y_{e,o_{N_{Gi}}}}{\Delta y_{e,o_{N_{Gi}}}} \right)^2 \geq 1 \end{cases} \tag{4.12}$$

where $\mathbf{s_{s,G_i}}$,$\mathbf{y_{e,s,G_i}}$ are the position vectors including the longitudinal and lateral positions [s_{s,G_i},y_{e,s,G_i}] sampled from the boundary of the hyper ellipse by using the parametric equations of a 4th order hyper ellipse:

$$s_{s,G_i} = s_{G_i} + \Delta s_{G_i} \sqrt{|\cos\theta|}\,\mathrm{sgn}(\cos\theta) \tag{4.13}$$

$$y_{e,s,G_i} = y_{e,G_i} + \Delta y_{e,G_i} \sqrt{|\sin\theta|}\,\mathrm{sgn}(\sin\theta) \tag{4.14}$$

where θ is a parameter sampled from $-\pi$ to π.

This optimization problem can be solved efficiently if good initial guesses are given. Figure 4-3 shows the execution time for solving the optimization problem under different numbers of OVs located randomly and sampling points on the hyper ellipse. All the problems are solved via active-set sequential quadratic programming (SQP) method in MATLAB Optimization Toolbox running in a laptop with Intel i5 4200U CPU, 2.4GHz and 4G RAM. It can be seen that with more object vehicles and finer sampling of the hyper

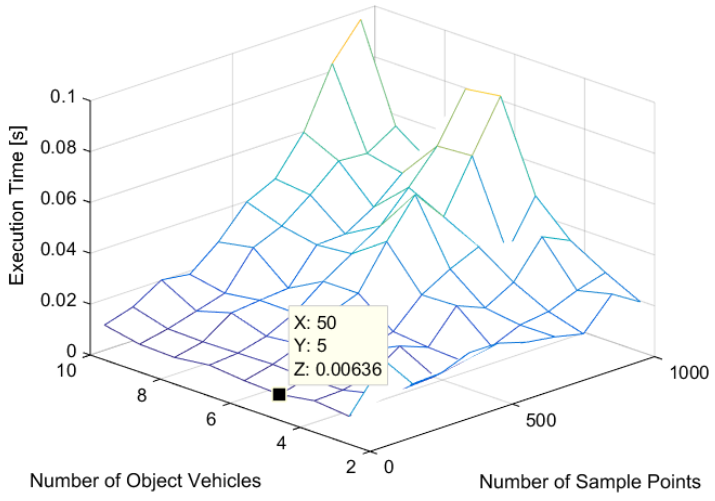


Figure 4-3: Estimated execution times for solving the optimization problem

ellipse, the execution times can be substantial (order of 40ms with 10 OVs and 500 samples). We can give an estimate of the complexity of solving these shape optimizations. Considering the number of predictive time steps N_p, and an interior-point solver (preferred for parallel computing implementation) which has a generally cubic complexity in the number of states N_x and constraints N_c, then the total computational complexity of determine one group shape for the whole predictive horizon is $O(N_p(N_x^3+N_c^3))$. Furthermore, multiple groups may be identified in one time step, as shown in Figure 3, which raises the complexity of determining the fields of multiple groups to at most $O(N_gN_p(N_x^3+N_c^3))$, where N_g is the number of identified groups. It should be noted that to have a good sampling of the boundary, N_c should usually be much larger than N_x (equal to just 2 for shape coordinates, say N_c=200 for a group with four object vehicles. Therefore, the approach require significant computational efforts. We had conjectured in [20] that GPU parallel computing could be a possible solution to accelerate the optimization, but we found in our tests since then that the achieved reduction is not substantial to realize real-time implementation. To overcome this issue and realize the benefits of the grouping scheme for real-time application, we integrate it with the following approach.

4.3.4 Supervised Learning Approach

As suggest in [98], a multilayer neural network (NN) with only one hidden layer can approximate a continuous function of *n* real variables arbitrarily well. Also, once it's trained, the complexity of calculating a NN function with one hidden layer is at most $O(N_n^2)$, where N_n is the number of the neurons in the hidden layer (N_n can be set to 10~25). Considering the length of predictive horizon and the multiple groups, the total complexity

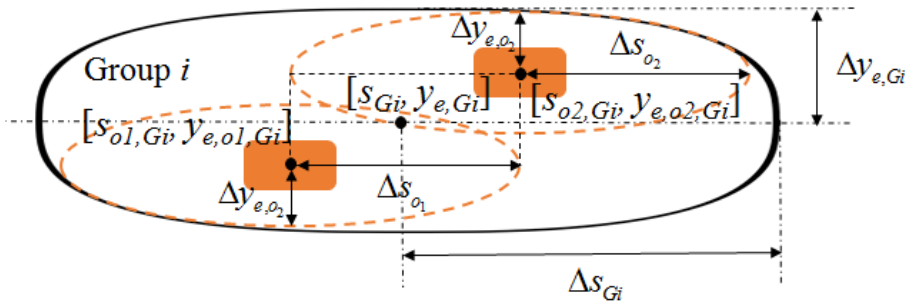


Figure 4-4: Example of relationship between the optimal $\Delta y_{e,G_i}$ and Δs_{G_i} and the parameters of the elliptical areas of the object vehicles

will rise to $O(N_g N_p\ N_n^2)$, which is much less than the optimization approach described above. Therefore, a NN based function can be trained to efficiently approximate the optimal $\Delta y_{e,G_i}$ and Δs_{G_i} in terms of the parameters of the elliptical areas of the object vehicles. This function is written compactly as:

$$\begin{bmatrix} \Delta s_{Gi} \\ \Delta y_{e,Gi} \end{bmatrix} = f\left(s_{o,Gi}, \Delta s_{o,Gi}, y_{e,o,Gi}, \Delta y_{e,o,Gi}\right) \tag{4.15}$$

where $\Delta s_{o,G_i}, \Delta y_{e,o,Gi}$ are the half major and half minor axle length sets of the elliptical areas of individual object vehicles in group *i*. See Figure 4-4 for an illustrative example.

Then, the pair training data required includes the input and outputs of the function. The half minor axes length $\Delta y_{e,o,Gi}$ for the regular-sized object vehicles are normally constant, thus it can be excluded in the inputs. For the half major axes length $\Delta s_{o,G_i}$, as it is proportional to the velocity of the ACV v_t, v_t can be used instead of $\Delta s_{o,G_i}$ as part of the inputs. Therefore, the function is rewritten as:

$$\begin{bmatrix} \Delta s_{Gi} \\ \Delta y_{e,Gi} \end{bmatrix} = f\left(s_{o,Gi}, y_{e,o,Gi}, v_t\right) \tag{4.16}$$

The training data for the function are generated by using the combination of the grid samples in the position space for s and y_e as well as the ACV velocity v_t (with the grid sizes denoted by N_s, N_y and N_v). Each sample from the grid will be used in the formulation of the optimization problem(4.11) to solve for the optimal $\Delta y_{e,G_i}$ and Δs_{G_i}, and used as the output training data of the function.

Applying combination theory, we can estimate the training data size requirements. A general NN function used to approximate the group shape for an arbitrary number (any group size/membership less or equal to) N_{OV} of object vehicles requires N_t pairs of samples of the input and output data pairs for function (4.16):

$$N_t = N_v \binom{N_s N_y}{N_{OV} - 1} = N_v \frac{\left(N_s N_y\right)!}{\left(N_{OV} - 1\right)!\left(N_s N_y - N_{OV} + 1\right)!} \tag{4.17}$$

Here, we consider that the grid coordinate system is fixed to on one of the object vehicle, then the required combinations is reduced from n_G to n_G-1, which helps to reduce the sample data size N_t.

Alternatively, one can also train a specific NN function for a specific number, i.e. N_{OV}, of object vehicles with the number of samples given by:

$$N_t = N_v \left[\binom{N_s N_y}{n_G - 1} - \binom{N_s N_y}{n_G - 2} \right] \tag{4.18}$$

Compared to the general NN function, the specific NN function have lower fitting complexity and higher accuracy, as it only covers the case with specific number of object

vehicles, thus requiring less training data. However, in the latter case, multiple specific NN functions will need to be trained independently to cover all possible cases when the number of object vehicles in the detection zone changes.

Once the training data are generated offline using a suitable optimization solver software/hardware, they can be used in the training process. Here, we use the neural network fitting toolbox in MATLAB to fit the parameters in the NN function. The performance of a trained general NN function is illustrated along with comparisons to optimization-based results in two examples cases with two and four object vehicle in the

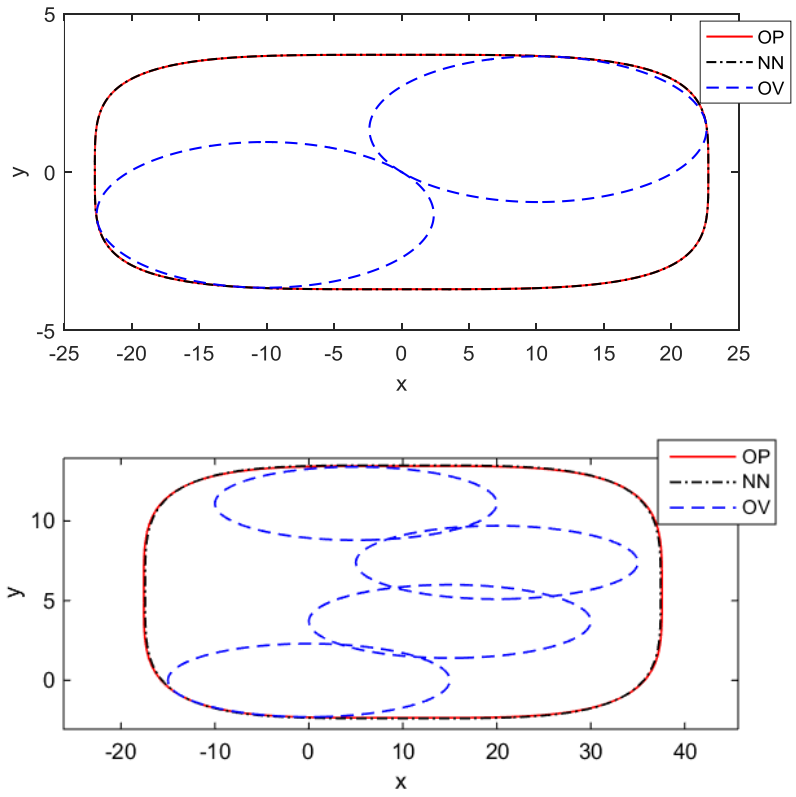


Figure 4-5: Illustration of the good approximation of the optimization results (OP) with a general NN function in the case of two and four object vehicles (OV)

group, as shown in Figure 4-5. In these example, the NN output is nearly indistinguishable from the optimization-based results.

4.4 Maneuver Planning and control Framework

We embed the above vehicle grouping algorithm within the constraint formulations for the maneuver planning and control framework described in our prior work [88]. The control framework is shown in Figure 4-6. The environment recognition module captures the environment information, such as lane marks, traffic signs or signals, the size or states of moving objects, the state of the ACV and its localization through camera, radar, lidar or wireless devices. The route navigator module works as a general GPS navigator, which plans the route from initial position to target destination via a map and localization of the

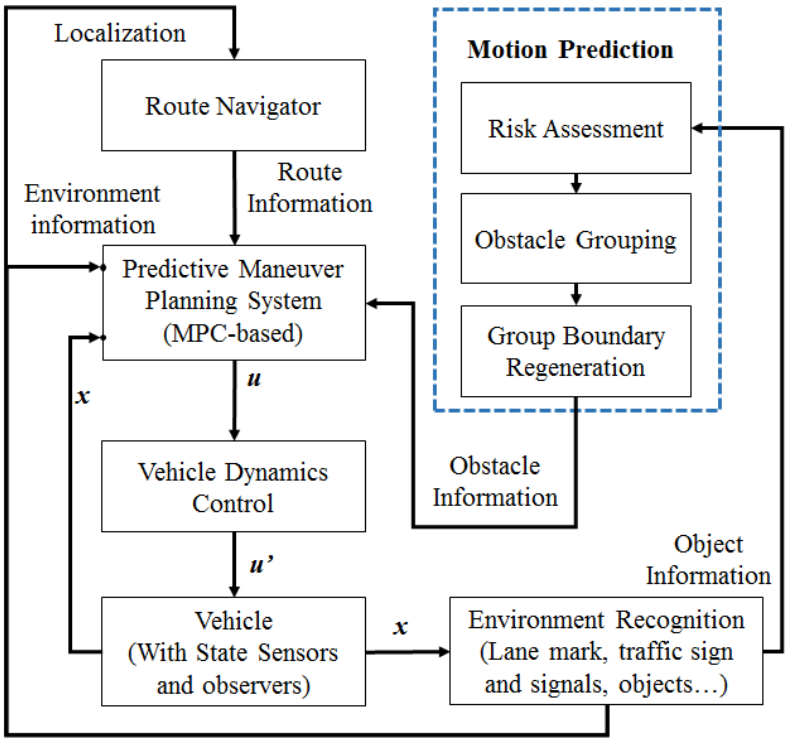


Figure 4-6: Control Framework

controlled vehicle. In the following discussions, we assume all the information from environment recognition and route navigator are known to the guidance system.

The predictive maneuver planning module is responsible for the maneuver and trajectory planning of the ACV. The multi-objective optimization problem solved at each MPC update are described by (3.16)~(3.21), Readers are referred to last chapter for a more detailed description.

4.5 Simulation and Results

In this section, we include some simulation results to illustrate the benefit of using the supervised learning based vehicle grouping algorithm. The interval of the sampling time of MPC is 0.15s and the prediction horizon is 6s.

Figure 4-7 shows part of the trajectory results of scenario 1 where the ACV is avoiding a group of slower object vehicles that are running at constant speed of 20m/s in

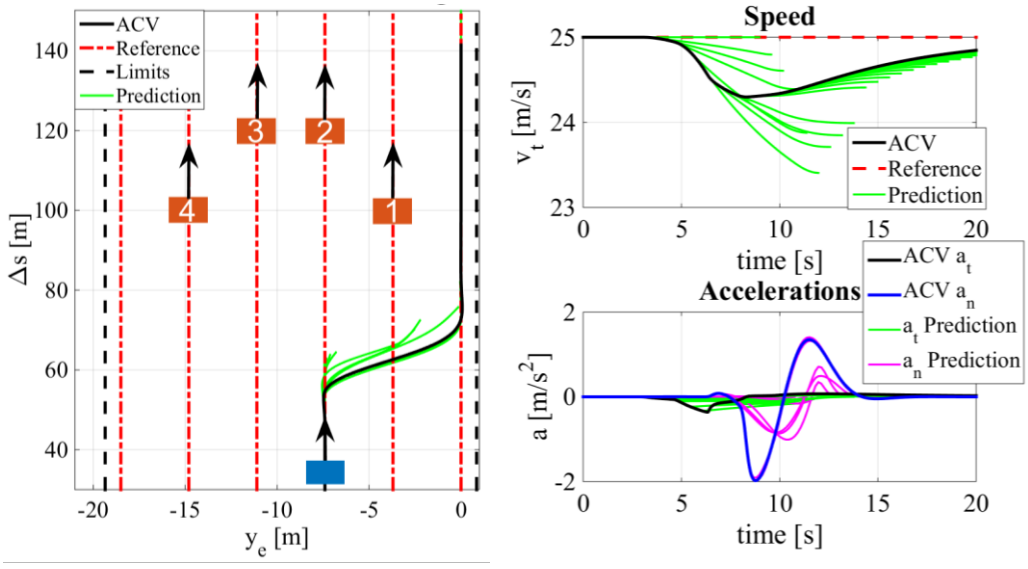


Figure 4-7: Results of state trajectories for scenario 1

the middle of the lane. The relative path profile is used to describe the position of the ACV according to the object vehicles. In [93], we showed that without the grouping algorithm, the ACV will be trapped in the local minimum created by the elliptical boundaries of the front object vehicles. However, with the grouping algorithm, the ACV easily avoids entering the field with local minimum.

Figure 4-8 presents the predicted trajectories for the simulation times at near 2s, 10s, 15s and 18s. At each simulation time, the predicted positions of the object vehicles as well as the related groups are illustrated by the samples taken at 1.5s (red), 3s (blue), 4.5s (green) and 6s (yellow). The blue line represents the predicted trajectory of the ACV. We can see that, initially(red), in the prediction horizon, the object vehicles are all in group 1

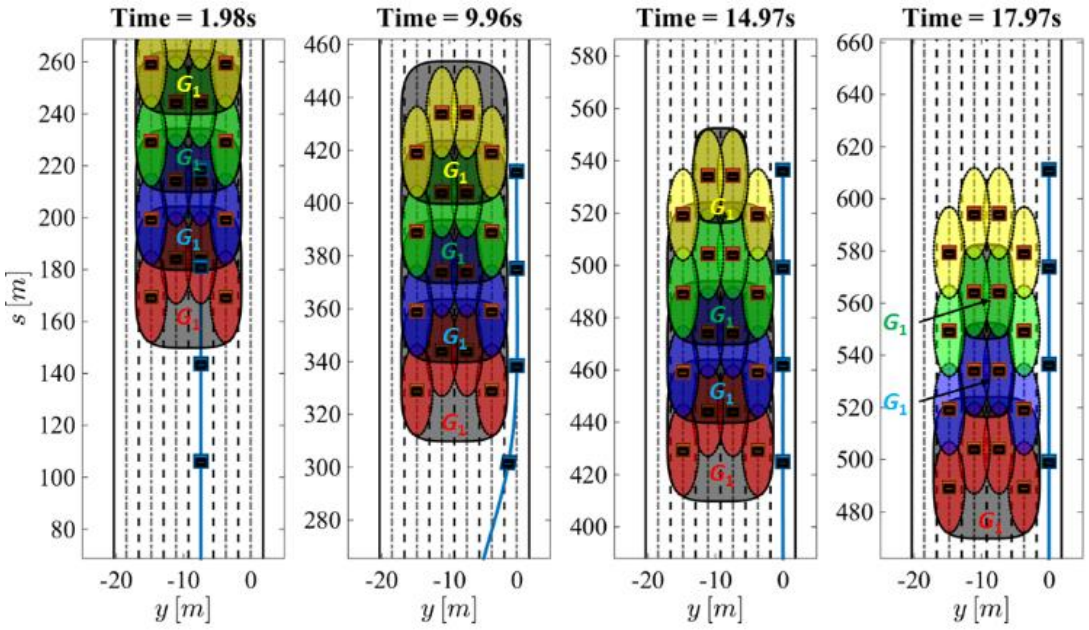


Figure 4-8: Predicted trajectories at different simulation times for scenario 1. In each case, 4 instants (at 1.5s, 3s, 4.5s and 6s) of the 6s prediction horizon, which are marked in blue for the ACV.

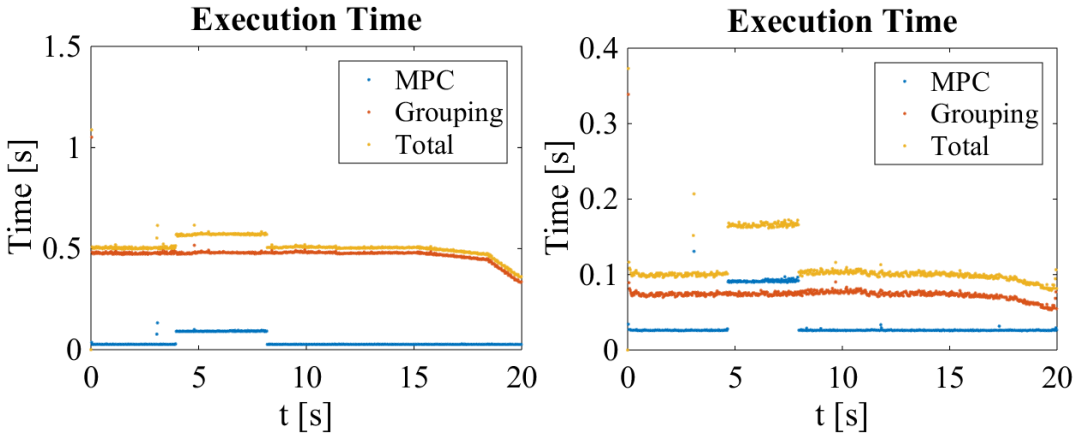


Figure 4-9: Execution time comparison of the predictive control system with optimization based(left) and NN based(right) grouping algorithm for scenario 1

(G_1) and as the ACV changes lane to pass object vehicles 1 and 4 (see Figure 4-7 for labels), the group splits and only vehicle 2 and 3 remain in G_1 (yellow). Finally, the group vanishes when the ACV passes all the object vehicles. The group behaviors in this scenario are all activated by the risk assessment of the object vehicles.

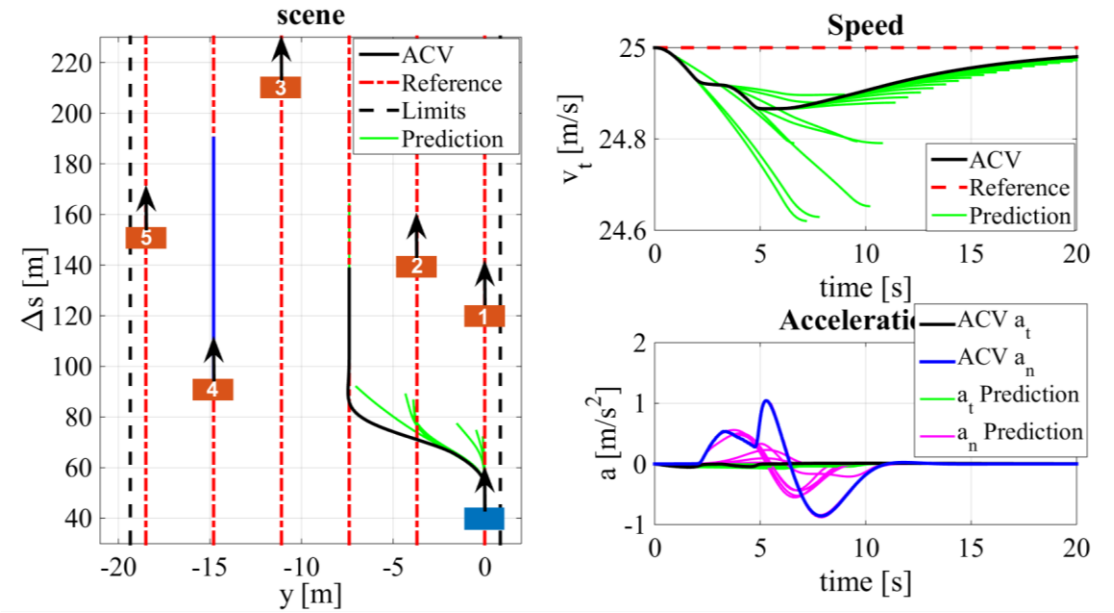


Figure 4-10: Results of state trajectories for scenario 2

Figure 4-9 shows the significant execution time reduction due to replacing the optimization based group parameter determination with NN functions. The maximum total time for executing one cycle of the grouping and MPC is around 0.15s, which is now close to the sampling time of the MPC. The step change of the execution time of MPC near 5 seconds and 8 seconds, is due to the engagement of the active constraint of the group in the MPC optimization problem when the ACV tries to pass the group.

The results of another scenario 2 is shown in Figure 4-10. In this case, object vehicle 4 passes object vehicle 5 and approaches object vehicle 3. The ACV changes lane from the most right lane to the middle to avoid the right side vehicle group G_1 consist of object vehicles 1 and 2. In the predictive trajectories shown in Figure 4-11, we can see the merging and splitting of G_2 when object 4 pass object 5. As for G_1, when the ACV passes the object 1, G_1 vanishes. But a (new) group G_1 forms again when object 4 is close to object 3. In the

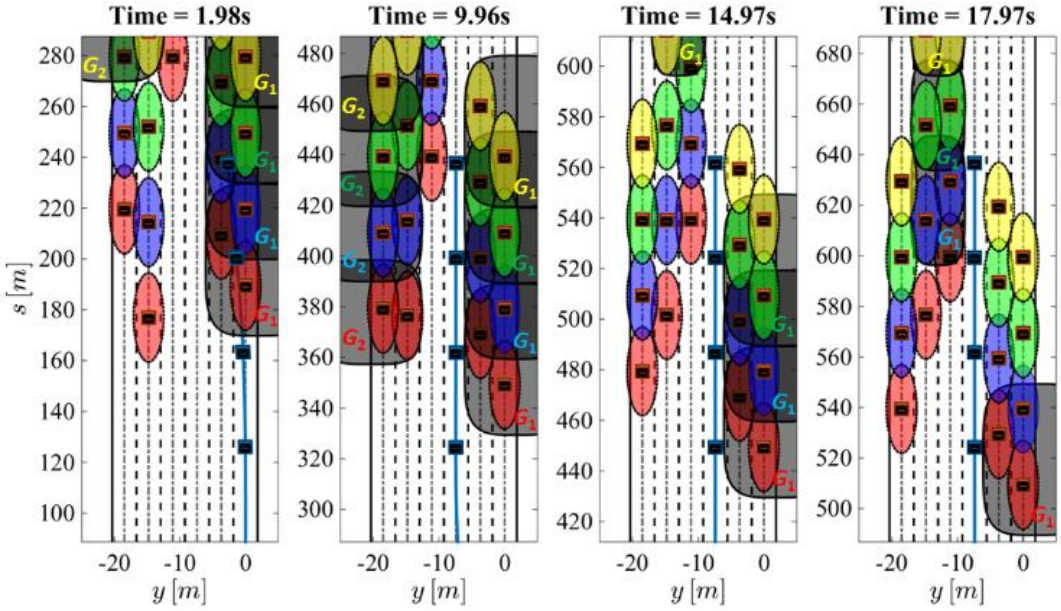


Figure 4-11: Predicted trajectories at different simulation times for scenario 2.

execution time comparisons, shown in Figure 4-12, the time primarily increases and decreases due to the group number change from 1 to 2 then back to 1. This is also reflected in the slope of the change in the execution time of the grouping algorithm.

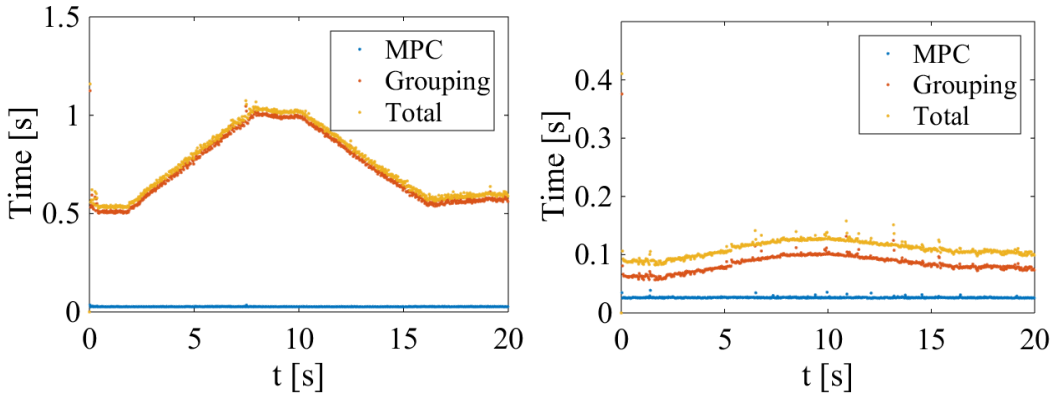


Figure 4-12: Execution time comparison of the predictive control system with optimization based(left) and NN based(right) grouping algorithm for scenario 2

4.6 Conclusion

In this chapter, we proposed a modeling framework for obstacle vehicle groups for use in a predictive control of an autonomous vehicle in highway public traffic. A systematic group modelling scheme is offered to describe the interaction between the object vehicles with merging and splitting behaviors. The proposed grouping algorithm can help to optimize the feasible field for the planning work by excluding the local minima created by the overlapping of the collision boundaries of multiple object vehicles. For online implementation, it is illustrated that a neural network(NN) function approximation can be used for group shape determination to obtain efficient, real time ready processing of the detected object vehicles and generating the related constraints for the MPC scheme used for motion planning and guidance.

Continuing work in this direction will consider the generally stochastic uncertainties in the object vehicle motion information and the resulting grouping model for predictive guidance, which will be introduced in the next chapter.

CHAPTER 5

A PROBABILISTIC FRAMEWORK FOR TRACKING THE FORMATION AND EVOLUTION OF MULTI-VEHICLE GROUPS

5.1 Abstract

Future self-driving cars and current ones with advanced driver assistance systems are expected to interact with other traffic participants, which often are multiple other vehicles. Object vehicle tracking forms a key part of resolving this interaction. Furthermore, descriptions of the vehicle group behaviors, like group formations or splits, can enhance the utility of the tracking information for further motion planning and control decisions. In this chapter, we propose a probabilistic method to estimate the formation and evolution, including splitting, re-grouping, etc., of object vehicle groups and the membership conditions for individual object vehicles forming the groups. A Bayesian estimation approach is used to first estimate the states of the individual vehicles in the presence of uncertainties due to sensor imperfections and other disturbances acting on the individual object vehicles. The closeness of the individual vehicles in both their positions and velocity is then evaluated by a probabilistic collision condition. Based on this, a density-based clustering approach is applied to identify the vehicle groups as well as the identity of the individual vehicles in each group. An estimation of the state of the group as well as of the group boundary is also given. Finally, detailed numerical experiments are included, including one on real-time traffic intersection data, to illustrate the workings and the

performance of the proposed approach. The potential application of the approach in motion planning of autonomous vehicles is also highlighted.

5.2 Introduction

In the march towards (semi-)autonomous driving, the task of guiding the controlled vehicle in the presence of other traffic participants remains a challenging problem. Therein, tracking of moving objects from sensor information plays a significant role. In particular, in public traffic, multiple other vehicles evolve in the traffic scene with changing velocity and positions. From the perspective of guidance and control of the individual autonomously controlled vehicle (ACV), group tracking can facilitate safe decisions and control actions for the current and upcoming maneuvers of the ACV. Group tracking entails the dynamic identification and estimation of group formation by merging attributes of individual objects, of the evolution of their motion as a group or multiple groups as well as the dissolution of groups by splitting [99], [100]. Group tracking information can constrain the nature of the interaction between the ACV and the other moving objects (primarily other vehicles in traffic).

For our purposes here, a group of objects is defined as a set of objects that have common movement (e.g. similar velocities) and close geometrical proximity. Depending on the ambiguity in the available measurement about the objects in the group, two categories of approaches to group tracking can be identified: (1) individual object-based approach [101] [102], and (2) extended object-based approach. In the first case, the measurement of the individual components in the group can be easily differentiated. In the second case, a too-close proximity between individual objects or overlapping sensor

information makes it hard to continuously distinguish individual objects. In the latter case, it's better to track the group as an extended object modeled with simple geometric shape like a circle [103], ellipse [104] [105], rectangle [105] [106] or some arbitrary shape [107] [108].

The extended object-based approach is usually used to identify the vehicle object from sets of measurements, e.g, a sparse laser point cloud. Data association approaches like Multi Hypothesis Tracking (MHT) [92], Probabilistic MHT (PMHT) [109], Probability Hypothesis Density (PHD) approach [110], Joint Probabilistic Data Association (JPDA) approach [111], or Random Finite Sets (RFS) [112] can be used to assign the measurements to each identified object vehicle. In addition, by considering the knowledge of the geometry of the object vehicle model, for example fused with camera images, the detected object vehicle can be represented by an extended object with an estimated spatial shape (center and extent parameters) and dynamics (location and velocity) [106], [113], [114].

For the individual object-based group tracking, interaction among the individual components of the group can be modelled by updating the group structure that results from behaviors including the occurrence or merging and splitting or vanishing of the group or groups. Two types of models have been used to describe the dynamic group structure: transition model [102] and evolution model [101]. In the transition model, specified Markov transition probabilities are used to represent the possible changes in the group structure. In the evolution model, the group association decisions are made based on the evaluation of the closeness between the objects within a group as well as the closeness

between the groups. The transition model allows a joint estimation of the group structure as well as the individual object states [102] [115], while the evolution model follows a hierarchical estimation pattern: first estimate the individual object states, then construct the group structure. The evolution model tracks the propagation of the closeness information which gives more clues about the potential inter-group and intra-group interactions and in general, does not require pre-specification of transition probabilities. While either approach entails more computational cost at implementation than individual object tracking, the hierarchical group tracking approach offers tractable formulations as we outline in this chapter. The obtained group structure information can subsequently simplify the motion planning problems for autonomous vehicles as we discuss below.

In our earlier work [93], we proposed a deterministic vehicle grouping method for groups of object vehicles that are then used for redefining the obstacle collision constraints for model predictive control (MPC) and guidance of an ACV. Therein, we formed groups between detected object vehicles based on a distance threshold defined by the overlap of their elliptical collision fields. The identified vehicle groups are then represented with the tightest/optimal hyper-elliptical boundaries. The results of our computational experiments showed that the vehicle group description with proper boundary design can redefine the feasible collision-free field to exclude undesired local minimums (for the motion plan) as it happens at the intersections of the collision boundaries of individual object vehicles. Later in [116], we refined the object vehicle grouping method with a group structure evolution model and applied a supervised learning method to reduce the on-line computational efforts of generating the optimal vehicle group boundaries. However, in

these previous works, uncertainties in the individual object vehicle (IOV) tracking due to sensor imperfections and environmental disturbances were not considered. Also, the closeness of the velocity of individual objects, which is an indicator of the similarity of their motion, was not used in the criteria for group formation.

In this chapter, we propose a probabilistic multiple vehicle grouping framework to track groups of IOVs with consideration of their finite geometric size information and closeness evaluation. This framework explicitly models uncertainty in the estimation of the states of IOVs and groups. The main contributions of this chapter are:

- Apply an evolution model to describe the update of object vehicle group (OVG) structure.
- Derive the probabilistic collision/closeness criteria between any two IOVs with non-negligible geometric size and shape information based on their state estimation via Bayesian tracking. A simplified derivation is also given for the case of Gaussian state distributions.
- Based on the closeness evaluation, a density-based method is applied to group/cluster the IOVs without a prior guess about the number of groups.
- The state of each OVG is determined by the weighted distribution of the state of each IOV in the OVG. The boundary of the OVG is calculated via approximation of the specific probability contours that consider the distribution of each IOV in the OVG.

5.3 Multiple Vehicle Grouping Framework Design

The object vehicle grouping framework follows a hierarchical estimation scheme to determine the group structure state $\boldsymbol{G}$ from the state of all detected object vehicles at time k. The overall framework is illustrated in Figure 5-1. The set denoted by $\boldsymbol{X}$ contains the states and geometrical shapes for all object vehicles. It is obtained by Bayesian IOV tracking given the measurement set $\boldsymbol{Z}$ from sensors. Then, a closeness matrix $\boldsymbol{M}_c$ calculated via probabilistic collision checking between each pair of object vehicles considering uncertainties and their finite geometrical sizes and shapes. Finally, a density-based clustering method (DBSCAN) with a threshold ε is used to group/cluster the IOVs and determine the group structure state $\boldsymbol{G}$. Each part of the framework is discussed in further detail in the following subsections.

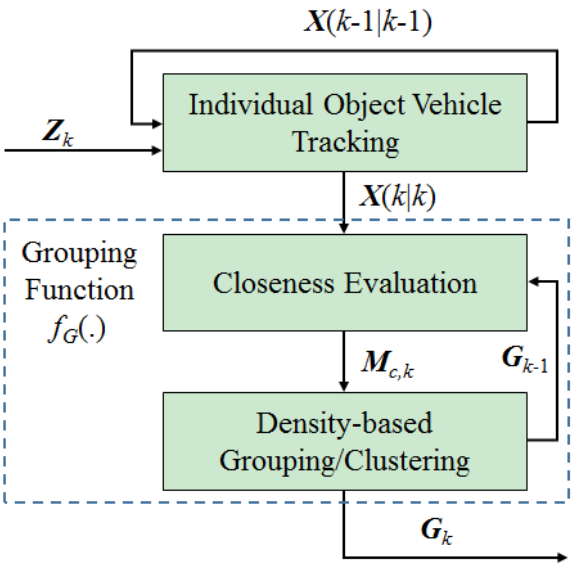


Figure 5-1: Object vehicle grouping framework

5.3.1 Modeling of Object Vehicle Group

Assuming there are IOVs indexed from 1 to N_{OV} (N_{OV}>1) that are being tracked at time k, the object vehicle group (OVG) structure state $\boldsymbol{G}_k$ is a collection of labeled vehicle groups:

$$\boldsymbol{G}_k = \left\{ G_{1,k}, ..., G_{N_G,k} \right\} \tag{5.1}$$

where N_G is the number of identified vehicle groups. A vehicle group i is defined as the tuple:

$$G_{i,k} = \left\langle x_{G,i,k}, S_{G,i,k}, I_{G,i,k}, B_{G,i,k} \right\rangle,\ i \in \left\{ 1, ..., N_{G,k} \right\} \tag{5.2}$$

where x_G is the state vector of the group that includes the estimated positions (of a representative point, e.g. centroid) and the velocities of the IOVs in the group as well as their covariances. S_G is a parameters set (or generally, an algebraic function $f_{s,G}$) that may be used to describe the current shape/contour of the group when considered as an extended rigid object. I_G is the index set of the IOVs that belongs to the group. All the components of the state vector X_G are determined by the states of the IOVs inside the group. B_G is the OVG behavior indicating the group structure change from the last time step, which would be one of the following three behaviors:

- Behavior 1: Merge. It happens when independent object vehicles or sub-groups merge in to the current group.
- Behavior 2: Split. It happens when a group is split into the current sub-group.
- Behavior 3: Continue. It happens when the group components stay the same.

Therefore, an evolution model of the object vehicle group structure is given by

$$\boldsymbol{G}_k = f_G\left(\boldsymbol{X}_k, \boldsymbol{G}_{k-1}\right) \tag{5.3}$$

where f_G is the grouping function that includes both the closeness evaluation and density-based grouping/clustering, see Figure 5-1. An example of the evolution of OVG structure is illustrated in Figure 5-2. Note that an individual group $G_{i,k}$ will be empty if there are no object vehicles inside it.

Behavior 1, or, respectively, Behavior 2, are usually activated by the condition that if the calculated probabilistic collision value (after closeness evaluation) between any two IOVs is higher, or respectively lower, than a threshold ε (see the dashed edge connecting the IOVs in Figure 5-2). The closeness evaluation is introduced in the Section 4.3.3 after we discuss the formulation for IOV tracking.

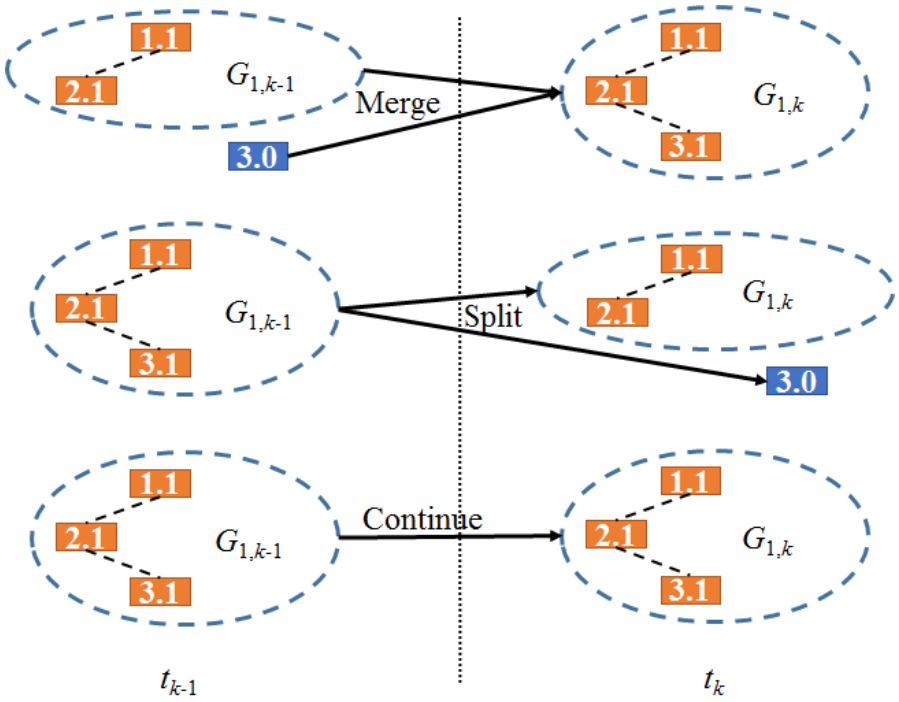


Figure 5-2: Illustration of object vehicle group behaviors (number before dot is vehicle index, number after dot is group index, 0 means no group)

5.3.2 IOV Tracking

For IOV tracking, we apply a Bayesian approach to estimate the motion states and the sizes (by object rigidity assumptions) of all the detected IOVs. By detected IOVs, we mean those falling in the range of the sensing system and deemed of interest for the tracking and guidance problem. Assuming there are IOVs indexed from 1 to N_{OV} (N_{OV}>1) being tracked at time k, the IOV set $\boldsymbol{X}$ is a collection of labeled IOV tuples:

$$\boldsymbol{X}_k = \left\{ X_{1,k}, ..., X_{N_{OV},k} \right\} \tag{5.4}$$

$$X_{j,k} = \left\langle x_{j,k}, S_{OV,j,k} \right\rangle,\ j \in \{1, ..., N_{OV}\} \tag{5.5}$$

where x is the estimate of the motion state vector of the IOV that includes the positions (a representative point, e.g. centroid) and the velocities of the object vehicle as well as their covariances. S_{OV} is the parameters set (or an algebraic function $f_{s,OV}$) used to describe the current shape/contour of the object vehicle, e.g. this could be the length and width for a rectangular description or the major and minor length for an elliptical description. Here, we assume S_{OV} and N_{OV} are already identified and we focus on the estimation of the motion state x. Methods to capture S_{OV}, N_{OV} can be found in [105], [106].

The general evolution of the motion state and measurement sequence of an IOV I can be written as:

$$x_{i,k} = f_{m,i}\left(x_{i,k-1}, w_{i,k-1}\right) \tag{5.6}$$

$$z_{i,k} = h_{m,i}\left(x_{i,k}, v_{i,k}\right) \tag{5.7}$$

where f_m is a (nonlinear) function of the state x and process disturbance/noise sequence w. z is the available measurement most likely from observation cameras, or on-board distance sensors (lidar). h_m is a (possibly nonlinear) function of the states x and measurement noise sequence v. The uncertainties considered in this chapter are mainly due to process noise from environmental disturbances like wind, road or un-modeled dynamics acting on the lateral and longitudinal motion of the vehicle and measurement/sensor noise. Later on, we will assume that these uncertainties are captured-well with Gaussian distributions.

In the Bayesian approach to tracking the motion state of an IOV i, one attempts to estimate the posterior probability density function (PDF) $p_i(x_{i,k}|z_{i,1:k})$ of the state x_i according to all the measurements z_i up to time k. Assuming the initial PDF $p_i(x_{i,0}|z_{i,0})\equiv p_i(x_{i,0})$ is available with no initial measurement z_0, then $p_i(x_{i,k}|z_{i,1:k})$ can be obtained by a two-step recursive loop: prediction and update.

In the prediction stage, if the required PDF $p_i(x_{k-1}|z_{1:k-1})$ is known, the prior PDF of the state at time k is predicted via the following equation:

$$p_i\left(x_{i,k} \mid z_{i,1:k-1}\right) = \int p_i\left(x_{i,k} \mid x_{i,k-1}\right) p_i\left(x_{i,k-1} \mid z_{i,1:k-1}\right) dx_i \tag{5.8}$$

Note that $p_i(x_{i,k}|x_{i,k-1})=p_i(x_{i,k}|x_{i,k-1}, z_{i,1:k-1})$ is obtained from equation (5.8) assuming a Markov process and the known statistics of $w_{i,k-1}$.

In the update stage, suppose a measurement $z_{i,k}$ is available, it can be used to correct the prior PDF via Baye's rule:

$$p_i\left(x_{i,k} \mid z_{i,1:k}\right) = \frac{p_i\left(z_{i,k} \mid x_{i,k}\right) p_i\left(x_{i,k} \mid z_{i,1:k-1}\right)}{p_i\left(z_{i,k} \mid z_{i,1:k-1}\right)} \tag{5.9}$$

Similarly, $p_i(z_{i,k}|x_{i,k})$ is obtained from equation (5.9) and the known statistics of $v_{i,k}$. The denominator term $p_i(z_{i,k}|z_{i,1:k\text{-}1})$ is given by:

$$p_i\left(z_{i,k} \mid z_{i,1:k-1}\right) = \int p_i\left(z_{i,k} \mid x_{i,k}\right) p_i\left(x_{i,k} \mid z_{i,1:k-1}\right) dx_i \tag{5.10}$$

Therefore, by following the recursive loop above, the posterior density of the motion state *x* for each IOV can be estimated. For a linear description of the motion and measurement system (3.2), the analytical solution for the exact posterior PDF can be obtained via the application of Kalman Filter (requiring Gaussian noise *v* and *w*,) and Grid-based Estimator (requiring discrete state space). For a nonlinear description of the system (3.1), Extended Kalman Filter or Unscented Kalman Filter, Approximate Grid-based Estimator or Particle Filter can be used to approximate the posterior PDF [117].

Without too much loss of generality, hereafter, *x* represents the motion state (both position and velocity components) for the centroid of the geometric shape of each object vehicle. We will also use *x* to refer to just the position component of the motion state (e.g. in illustrations), when there is no ambiguity.

5.3.3 Closeness Evaluation

As the state of the IOVs are estimated by the posterior PDF for the centroid of each vehicle (including uncertainty), the Euclidean distance metric is not suitable to represent the closeness between different IOVs. For such a case, the probability of collision between two IOVs can be applied to measure closeness. Let $\mathbb{X}_i(x_{i,k})$ be the state space (position,

velocity and shape) occupied by IOV i at time k considering its geometric shape, e.g. an area described by an algebraic function $f_{s,OV,i}(x)$. Then, collision between IOV i and IOV j is defined by the condition $C(x_{i,k}, x_{j,k})$: $\mathbb{X}_{i,k}(x_{i,k}) \cap \mathbb{X}_{j,k}(x_{j,k}) \neq \emptyset$. Then, the probability of collision between the two IOVs is defined by the integral of the joint state distribution of the IOV i and IOV j:

$$P_C\left(x_{i,k}, x_{j,k}\right) = \int\int I_C\left(x_{i,k}, x_{j,k}\right) p_{ij}\left(x_{i,k}, x_{j,k}\right) dx_i dx_j \tag{5.11}$$

where I_C is the collision indicator function defined by:

$$I_C\left(x_{i,k}, x_{j,k}\right) = \begin{cases} 1, \text{ if } \mathbb{X}_{i,k}\left(x_{i,k}\right) \cap \mathbb{X}_{j,k}\left(x_{j,k}\right) \neq \emptyset \\ 0, \text{ otherwise} \end{cases} \tag{5.12}$$

This formulation of probability of collision can be implemented via Monte Carlo Simulations (MCS), which are computationally expensive. With assumptions of Gaussian distributions, an approximate closed-form solution was given in [118] for pairs of small sized objects with one of which can be reduced to a point. Then, the PDF value of the x in $\mathbb{X}(x)$ are nearly the same as the one in the centroid of $\mathbb{X}(x)$. However, in our case, the sizes of the IOVs are not negligible and such approximations will not work. Therefore, we develop some strategies to approximate the probability of collision between two IOVs with non-negligible geometric sizes and shapes.

The first step is to approximate the collision indicator function with a simpler description. Here, we rewrite the definition of the collision condition of two IOVs (with indices i and j) with non-negligible geometric sizes at time k as $C'(x_{i,k}, x_{j,k})$: $x_{i,k} \in \mathbb{X}_{ij,k}(x_{j,k})$, where $\mathbb{X}_{ij,k}(x_{j,k})$ is an extended deterministic geometric space occupied by IOV j at time k

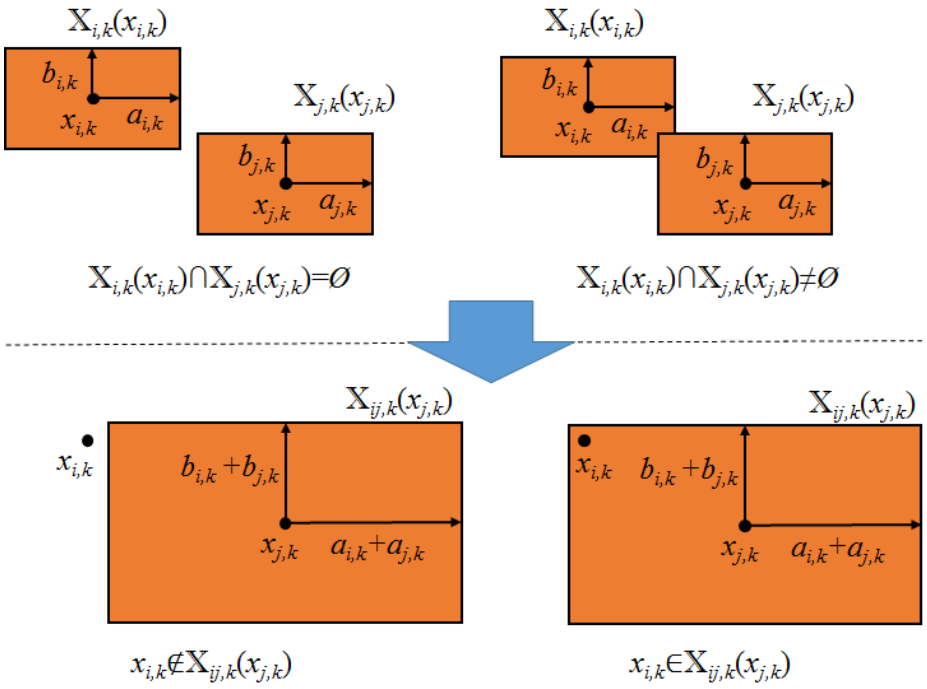


Figure 5-3: Example of the collision condition for two IOVs with rectangular shape description in 2D (*a* is half length and *b* is half width)

on which we lump the geometric shapes/sizes of *both* IOVs (with indices *i* and *j*). Therein, IOV *i* is considered as a point. An example of collision in 2D position space between two IOVs with rectangular shapes is shown in Figure 5-3. One can also similarly derive the extended shape $\mathbb{X}_{ij,k}(x_{j,k})$ for other geometric descriptions like circles or ellipses [93]. Then, the collision condition can be represented by an inequality in terms of the relative distance between $x_{i,k}$ and $x_{j,k}$. The collision indicator function can be rewritten as:

$$I_C\left(x_{i,k}, x_{j,k}\right) = \begin{cases} 1, \text{ if } x_{i,k} \in \mathbb{X}_{ij,k}\left(x_{j,k}\right) \\ 0, \text{ otherwise} \end{cases} \tag{5.13}$$

Therefore, (5.11) can be modified as:

$$P_C\left(x_{i,k}, x_{j,k}\right) = \int \left[\int_{x_{i,k} \in \mathbb{X}_{ij,k}\left(x_{j,k}\right)} p\left(x_{i,k} \mid x_{j,k}\right) dx_i \right] p_j\left(x_{j,k}\right) dx_j \qquad (5.14)$$

As the inner integral of (5.14) constrains the range of $x_{i,k}$ within $\mathbb{X}_{ij,k}(x_{j,k})$, we can define a deviation state variable $\Delta x_{j,k} \in \mathbb{X}_{ij,k}(0)$ to replace $x_{i,k}$:

$$\int_{x_{i,k} \in \mathbb{X}_{ij,k}\left(x_{j,k}\right)} p\left(x_{i,k} \mid x_{j,k}\right) dx_i = \int_{\Delta x_{j,k} \in \mathbb{X}_{ij,k}(0)} p\left(x_{i,k} = x_{j,k} + \Delta x_{j,k} \mid x_{j,k}\right) d\Delta x_j \qquad (5.15)$$

where $\mathbb{X}_{ij,k}(0)$ is the lumped space when $x_{j,k}$ is at the origin. Then,

$$P_C\left(x_{i,k}, x_{j,k}\right) = \int \left[\int_{\Delta x_{j,k} \in \mathbb{X}_{ij,k}(0)} p\left(x_{i,k} = x_{j,k} + \Delta x_{j,k} \mid x_{j,k}\right) d\Delta x_j \right] p_j\left(x_{j,k}\right) dx_j \qquad (5.16)$$

This integral can be further simplified when the distributions of the states of IOV *i* and IOV *j* are Gaussian and independent.

Proposition 1: Consider IOV *i*, with a point description with state $x_{i,k}$ and IOV *j* with state $x_{j,k}$ and an extended deterministic geometry description $\mathbb{X}_{ij,k}(x_{j,k})$). If the states $x_{i,k}$ and $x_{j,k}$ have Gaussian distributions, i.e., $x_{i,k} \sim N(\mathrm{m}_{i,k}, \Sigma_{i,k})$, $x_{j,k} \sim N(\mathrm{m}_{j,k}, \Sigma_{j,k})$, and the state tracks of IOV *i* and IOV *j* are independent, then:

$$P_C\left(x_{i,k}, x_{j,k}\right) =$$

$$\int_{\Delta x_{j,k} \in \mathbb{X}_{ij,k}(0)} \frac{\exp\left[-\frac{1}{2}\left(\Delta x_{j,k} + \mathrm{m}_{i,k} - \mathrm{m}_{j,k}\right)^T \left(\Sigma_{i,k} + \Sigma_{j,k}\right)^{-1} \left(\Delta x_{j,k} + \mathrm{m}_{i,k} - \mathrm{m}_{j,k}\right) \right]}{\sqrt{(2\pi)^{n_x} \left|\Sigma_{i,k} + \Sigma_{j,k}\right|}} d\Delta x_j \qquad (5.17)$$

where n_x is the dimension of the state *x* (generally comprising of the position and the velocity for each IOV).

Poof: Let $p_x(\mathrm{m}, \Sigma)$ denote the PDF of a multivariate Gaussian distribution for IOV state x. Given the independence assumption:

$$\begin{aligned}&\int\left[\int_{\Delta x_{j,k}\in \mathbb{X}_{ij,k}(0)} p\left(x_{i,k}=x_{j,k}+\Delta x_{j,k}\mid x_{j,k}\right)d\Delta x_j\right]p\left(x_{j,k}\right)dx_j\\&=\int_{\Delta x_{j,k}\in \mathbb{X}_{ij,k}(0)}\left[\int p_{x,j}\left(\mathrm{m}_{i,k}+\Delta x_{j,k},\Sigma_{i,k}\right)p_{x,j}\left(\mathrm{m}_{j,k},\Sigma_{j,k}\right)dx_j\right]d\Delta x_j\end{aligned} \tag{5.18}$$

Using the fact that the product of two multivariate Gaussian distributions is also a multivariate Gaussian [119]:

$$p_{x,j}\left(\mathrm{m}_{i,k}+\Delta x_{j,k},\Sigma_{i,k}\right)p_{x,j}\left(\mathrm{m}_{j,k},\Sigma_{j,k}\right)=c\cdot p_{x,j}\left(\mathrm{m}_{c,k},\Sigma_{c,k}\right) \tag{5.19}$$

where:

$$c=\frac{\exp\left[-\frac{1}{2}\left(\mathrm{m}_{i,k}+\Delta x_{j,k}-\mathrm{m}_{j,k}\right)^T\left(\Sigma_{i,k}+\Sigma_{j,k}\right)^{-1}\left(\mathrm{m}_{i,k}+\Delta x_{j,k}-\mathrm{m}_{j,k}\right)\right]}{\sqrt{(2\pi)^{n_x}\left|\Sigma_{i,k}+\Sigma_{j,k}\right|}} \tag{5.20}$$

$$\mathrm{m}_{c,k}=\left(\Sigma_{i,k}^{-1}+\Sigma_{j,k}^{-1}\right)^{-1}\left(\Sigma_{i,k}^{-1}\mathrm{m}_{i,k}+\Sigma_{j,k}^{-1}\mathrm{m}_{j,k}\right) \tag{5.21}$$

$$\Sigma_{c,k}=\left(\Sigma_{i,k}^{-1}+\Sigma_{j,k}^{-1}\right)^{-1} \tag{5.22}$$

Using (5.19)-(5.22) in equation (5.18), we have:

$$\begin{aligned}&\int\left[\int_{\Delta x_{j,k}\in \mathbb{X}_{ij,k}(0)} p\left(x_{i,k}=x_{j,k}+\Delta x_{j,k}\mid x_{j,k}\right)d\Delta x_j\right]p_j\left(x_{j,k}\right)dx_j\\&=\int_{\Delta x_{j,k}\in \mathbb{X}_{ij,k}(0)}\left[c\cdot\int p_{x,j}\left(\mathrm{m}_{c,k},\Sigma_{c,k}\right)dx_{j,k}\right]d\Delta x_j\end{aligned} \tag{5.23}$$

As the inner integral equals to 1, equation (5.17) is proved.

Remark 5-1: By following the redefinition of the collision condition and *Proposition 1*, we can see that the problem of evaluating the probability of collision between two IOVs with non-negligible geometric sizes/shapes with Gaussian distributed and independent states can be transformed into one of calculating the integral of a combined multivariate Gaussian density function (defined by equation (5.20)) within a specified integral space (defined by $\mathbb{X}_{ij,k}(0)$). Furthermore, if the combined covariance $\Sigma_{i,k}+\Sigma_{j,k}$ is diagonal and $\mathbb{X}_{ij,k}(0)$ is a combination of closed integral ranges for each variate of $\Delta x_{j,k}$, a closed-form solution can be found for the probability of collision by evaluating (5.17).

Remark 5-2: For evaluation of closeness between IOVs via the collision probability, the state variates are taken from the IOV tracking. The closeness here includes not only the "nearness" in positions but also the "similarities" in velocities between the IOVs. Similar to the rectangles used to illustrate the closeness in positons (in Figure 5-3 and discussions above), a speed range can also be used to define the closeness in velocities. With x interpreted as the motion state vector (position and velocity), both aspects of closeness are already considered above, including in the specification of the integral space $\mathbb{X}_{ij,k}(0)$. Only if the nearness in both positions and velocities are satisfied are any two IOVs considered close to each other.

If the Gaussian distributed motion states of IOV i and IOV j are dependent as is possible for cases with mutual interactions, a different result that approximates the collision probability equation (5.16) may be sought. However, we do not address these cases in this chapter. Some discussions in this direction can be found in [118].

Furthermore, for the case of non-Gaussian distributed motion states of IOV *i* and IOV *j*, whether these are dependent or not, one may have to resort to MCSs to evaluate the probability of collision directly from (5.16).

Finally, by applying the probabilistic closeness/collision evaluation between each pair of detected IOVs, a closeness *matrix* $\boldsymbol{M}_c$ can be assembled:

$$\boldsymbol{M}_C = \begin{bmatrix} P_C\left(x_{1,k}, x_{1,k}\right) & \cdots & P_C\left(x_{1,k}, x_{N_{OV},k}\right) \\ \vdots & \ddots & \vdots \\ P_C\left(x_{N_{OV},k}, x_{1,k}\right) & \cdots & P_C\left(x_{N_{OV},k}, x_{N_{OV},k}\right) \end{bmatrix} \tag{5.24}$$

This matrix will be used in the grouping of IOVs in the next sub-section.

5.3.4 Density-Based Grouping/Clustering

Here, we adopt the Density-based Spatial Clustering of Applications with Noise (DBSCAN) approach [120] to group the detected IOVs by processing the closeness matrix given by equation (24). The probability of collision (value between 0 and 1) between the pairs of IOVs provides a good one-dimensional closeness indicator that be used with DBSCAN [121]. DBSCAN is widely applied in machine learning and data mining for clustering purposes due to its attractive attributes:

- The number of clusters/groups in the data set is not required to be pre-specified.
- Only two parameters are required: the closeness threshold ε in the neighborhood of any object *i* and the minimum number of other objects μ that are within the threshold ε of object *i*.
- It is suitable for arbitrarily shaped clusters/groups.

The key idea we adopt from this approach for vehicle grouping is that for the main object components of an OVG, at least a minimum number μ of IOVs should be contained in the neighborhood of a given closeness threshold ε (between 0% and 100%). Here, we emphasize that selecting an appropriate ε is very important as it's directly related to the success of the grouping algorithm. For example, a too small ε (≈0%) may lead to big groups with low-closeness IOVs inside, and a too large ε (≈0%) may fail in grouping the IOVs even for those with high closeness. A proper range for ε could depend on the situation (urban, highway, intersection, etc). Here, we select ε to be 0.5 (average of the collision probability of 0% and 100%) for our illustrations. The selection of μ depends on the density of the objects. These main object components in the group are defined as core object vehicles (COVs). The closeness threshold ε defines a *density connection condition* between two IOVs: if $P_C(x_{i,k}, x_{j,k}) \geq \varepsilon$, IOV i and IOV j are said to be density connected with each other. There can also be another kind of IOV called border object vehicle (BOV) in the

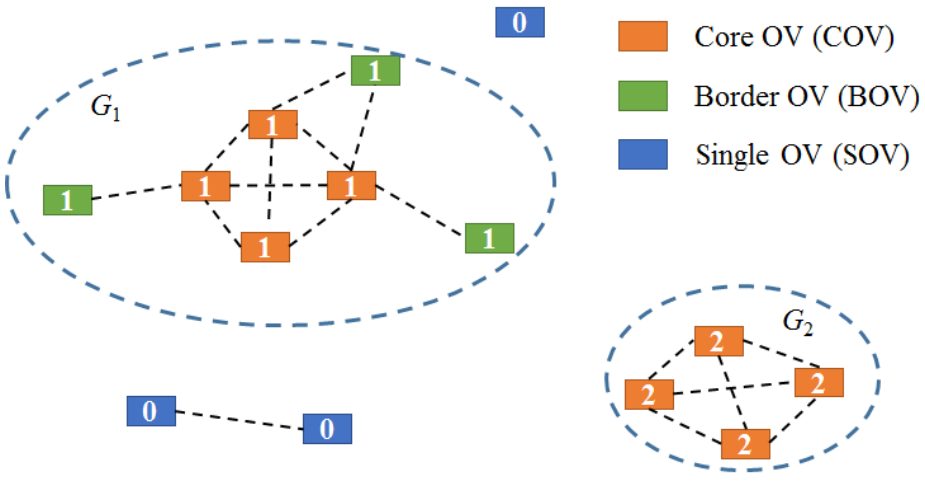


Figure 5-4: Illustration of the DBSCAN grouping results (μ=4). The edge means there is density-connection between the IOVs. Only the group index of each IOV is shown here. 0 means the IOV is SOV with no group index.

group that can't satisfy the minimum number μ requirement for being a COV but can be connected with COVs. In addition, there can be IOVs not connected with any COVs. These are considered as single object vehicles (SOVs). An illustration of these different kinds of objects is given in Figure 5-4. The algorithm is detailed in [120].

Once the clustering/grouping is done, each IOV will be labeled with its updated OVG index from 0 to N_G. And all the indices of the IOVs in OVG i will be stored in an index set as $I_{G,i}$. Furthermore, the group behavior $B_{G,i}$ can also be determined by evaluating the OVG index for each IOV at sequential time steps.

Although the DBSCAN approach is only applied here to identify the OVGs and label the IOVs with their OVG index, the motion state for OVG i $x_{G,i}$ at time k can be obtained from the mixed state distribution of those IOVs in the group:

$$x_{G,i,k} = \sum_{j=1}^{N_{I_{G,i,k}}} \omega_{G,i,j,k} \int p_{I_{G,i,k}(j)} \left(x_{I_{G,i,k}(j),k} \right) dx_{I_{G,i,k}(j)} \tag{5.25}$$

where $N_{IG,i,k}$ is the number of elements in the index set $I_{G,i}$ at time k. The weights for different IOV distributions can be determined based on the closeness of each IOV to other IOVs in the same OVG:

$$\omega_{j,k} = \frac{\sum_{i \in I_{G,i,k} \setminus I_{G,i,k}(j)} P_C \left(x_{I_{G,i,k}(j),k}, x_{i,k} \right)}{\sum_{j=1}^{N_{I_{G,i,k}}} \sum_{i \in I_{G,i,k} \setminus I_{G,i,k}(j)} P_C \left(x_{I_{G,i,k}(j),k}, x_{i,k} \right)} \tag{5.26}$$

Finally, the shape of the OVG *i* at time *k* can be described by the boundary of an area with a specified joint probability distribution among all the IOVs in the OVG *i*, i.e. the set:

$$\left\{x_k : P_{G,i}\left(x_k\right) = \alpha,\ 0 < \alpha < 1\right\} \tag{5.27}$$

$$P_{G,i}\left(x_k\right) = P\left(\bigcup_{j \in I_{G,i,k}} x_{j,k} = x_k\right) \tag{5.28}$$

According to the inclusion-exclusion principle of set theory, equation (5.28) becomes:

$$\begin{aligned} P\left(\bigcup_{j \in I_{G,j,k}} x_{j,k} = x_k\right) &= \sum_{j=1}^{N_{I_G,i,k}} P_j\left(x_k\right) - \sum_{\substack{J \subseteq I_{G,i,k} \\ |J|=2}} P\left(\bigcap_{j \in J} x_{j,k} = x_k\right) \\ &+ \sum_{\substack{J \subseteq I_{G,i,k} \\ |J|=3}} P\left(\bigcap_{j \in J} x_{j,k} = x_k\right) - \cdots + (-1)^{N_{I_G,i,k}} P\left(\bigcap_{j \in I_{G,i,k}} x_{j,k} = x_k\right) \end{aligned} \tag{5.29}$$

As calculating the intersection distribution probability among the IOVs in the OVG requires multiple integrals (with the order equal to the number of IOVs in the group), it's hard to evaluate the probability of collision via equation (5.29), especially when $N_{IG,i,k}$ had a large value. Therefore, we need a tractable approximation to (5.29). If we ignore the intersection probability calculation, we obtain a conservative evaluation of the collision probability:

$$P_{G,i}\left(x_k\right) \approx \sum_{j=1}^{N_{I_G,i,k}} P_j\left(x_k\right) \tag{5.30}$$

We say equation (5.30) is a conservative evaluation because the probability is overestimated by simply adding the probabilities based on the distribution of each IVO. This is known as the Boole's inequality or union bound [122]:

$$P\left(\bigcup_{j\in I_{G,j,k}} x_{j,k} = x_k \right) \le \sum_{j=1}^{N_{I_G,i,k}} P_j\left(x_k\right) \tag{5.31}$$

The OVG boundary is then obtained by drawing the probability contour using numerical methods:

$$\left\{ x_k : \sum_{j=1}^{N_{I_G,i,k}} P_j\left(x_k\right) = \alpha,\ 0<\alpha<1 \right\} \tag{5.32}$$

A comparison of the OVG boundaries determined by equation (5.29) and (5.30) is illustrated in the example in Figure 5-5. We can see the probability of the group distribution is bounded by (5.30).

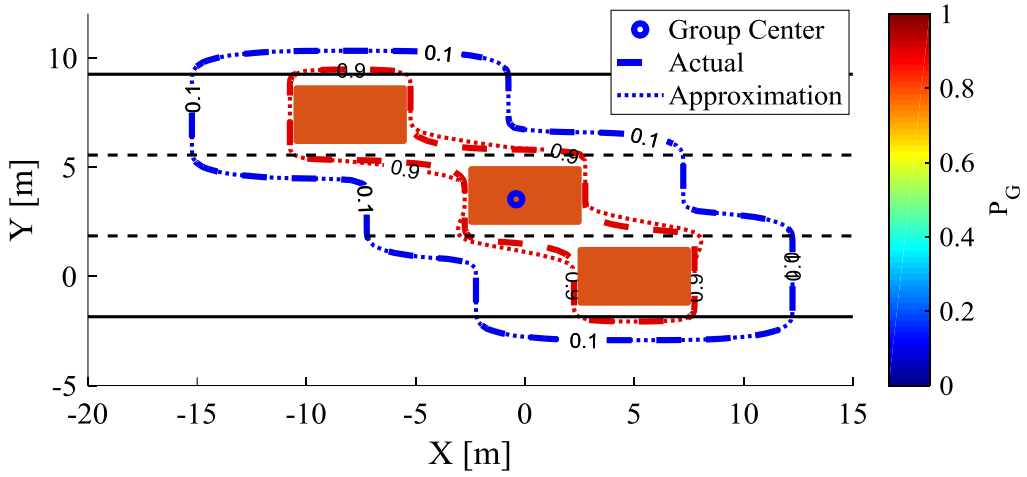


Figure 5-5: Illustration of the OVG distribution contour in a 2D position space. The position states of the three IOVs are assumed to be Gaussian distributed. α=0.1 and 0.9 contours shown. Solid probability contours are calculated by (5.29) while the dash contours come from (5.30).

5.4 Numerical Experiment and Discussion

To illustrate the performance of the proposed object vehicle grouping framework, we first include the setup and results of a numerical experiment that represents a complex highway scenario. Several methods are compared for use in the closeness evaluation and the salient aspects of the group tracking approach are illustrated with this scenario. We then present the results of the application of the proposed approach to a real-time traffic intersection scenario from the Next Generation Simulation (NGSIM) project database available on the Research Data Exchange of the U.S. Department of Transportation's Federal Highway Administration [123]. In both scenarios, we assume a centralized surveillance view of the IOVs from the ego-vehicle or roadside infrastructure, and illustrate the performance of the vehicle grouping algorithm.

5.4.1 Complex Highway Scenario

In the highway scenario, we use a linear kinematic particle motion model defined in the Frenet frame with KF for IOV state tracking purposes. This model has been used in our previous chapter for motion planning purposes. See (3.2) in section 3.3.1 for details. Here, we assume that the measurements are obtained without sensor delay, faults or sensing range limitations. With such linear dynamics models for the IOVs, a regular KF can be used to estimate the states of the IOVs. Note that even more refined implementations such as Interactive Multi-Model KF [80] and higher order models are also possible to use for IOV state estimation and integrated with our grouping function/approach as depicted in Figure 5-1.

As for the closeness calculation, we consider both the closeness in positions and velocity in the two scenarios. To calculate the closeness in positions, the geometric shape of the IOV is defined as a rectangle with a car-like realistic size (a for half length and b for half width). Also, in the numerical experiment, we will add a safety margin $c_s v^s_{t,o,i}$ that is related to the velocity of the IOV i in its geometric length to mimic human-driver like actions that keep a safe distance between a front and rear vehicle, as shown in Figure 5-6.

For the closeness in velocity, we define a bound $[-\Delta v_t, \Delta v_t]$ for the velocity difference between two IOVs. This will factor in following and leading conditions in the probabilistic inclusion/exclusion of IOVs in groups. After the closeness evaluation, the OVGs will be identified via DBSCAN w.r.t the closeness threshold ε and the minimum

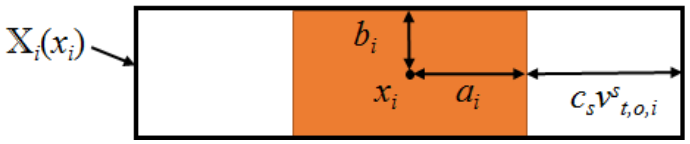


Figure 5-6: Geometric shape of the IOV i in the numerical simulation (c_s is a constant time gap to adjust the safety margin)

Table 5-1: Parameters of IOVs in the numerical experiment

Parameter	Value	Parameter	Value	Parameter	Value
$w_{\dot{s},o}$ [m/s]	$N(0,25)$	K_{s2}	2	c_s [s]	0.5
$w_{y,o}$ [m]	$N(0,1)$	K_{y1}	2.5	Δv_t [m/s]	1
$v_{s,o}$ [m]	$N(0,25)$	K_{y2}	2	ε	0.5
$v_{y,o}$ [m]	$N(0,1)$	a [m]	3.75	μ	2
K_{s1}	2.5	b [m]	1.6	α	0.5

number μ. Here we choose μ=2 due to the small numbers of IOVs (low density) in the present example, which considers 8 IOVs on a highway scenario (described below). Therefore, in this case, the BOV and COV are the same. All the parameters used in the numerical experiment are given in Table 5-1.

The highway scenario we constructed is a sequence of typical highway situations, like cruising, overtaking, following etc, are specifically selected to illustrate the nuances of the group evolution for a span of 90 seconds. The reader is encouraged to look at the

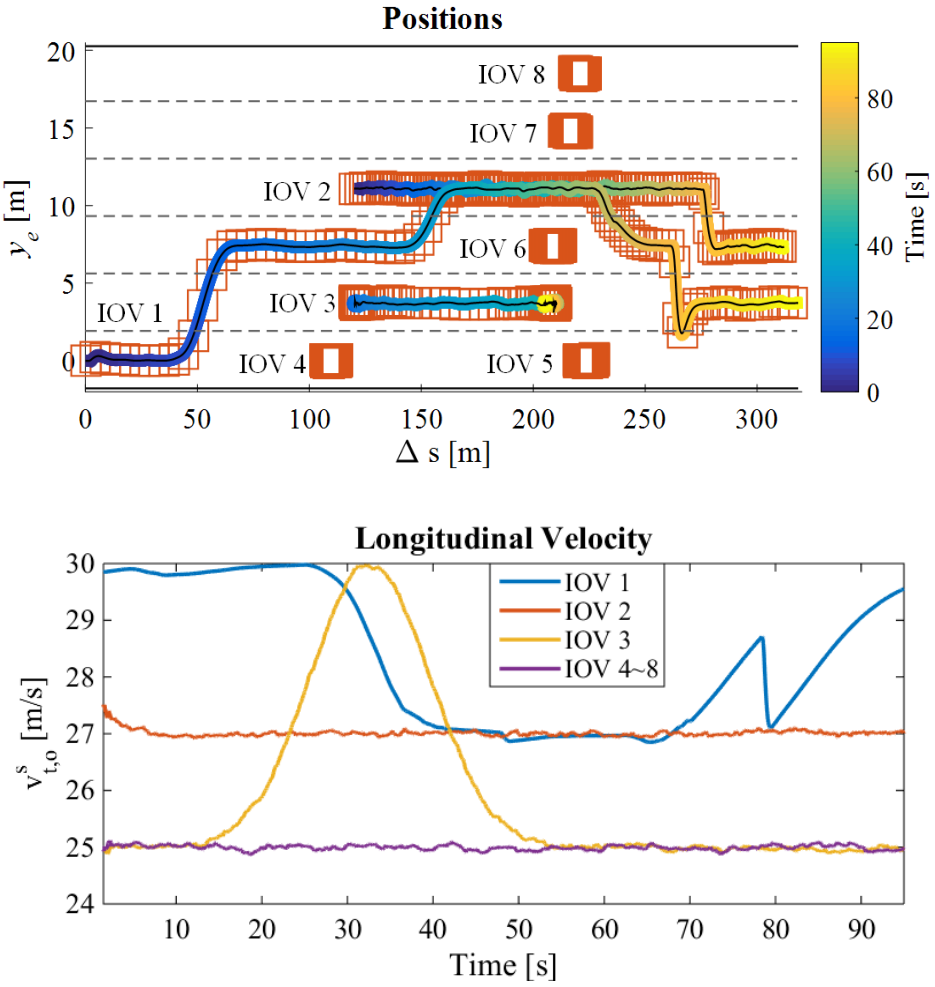


Figure 5-7: States of the IOVs in a highway scenario. Top: relative positions, Bottom longitudinal velocities.

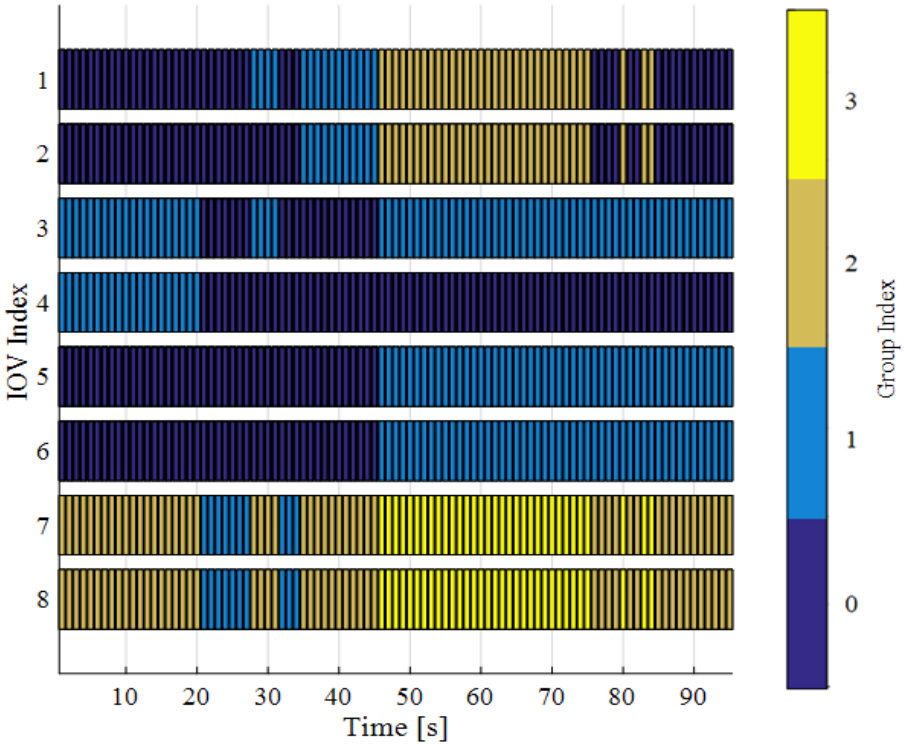


Figure 5-8: Group structure evolution for the highway scenario with application of either the NI or MSC methods for closeness evaluation.

state tracking (estimation) results for all IOVs shown in Figure 5-7 at this point. These are elaborated further in the next subsection.

We compare our proposed approach to closeness evaluation and grouping in the highway scenario, we compare our numerical integration (NI) method on the derived condition (5.17) with the Monte Carlo Simulations (MCS) using 100000 samples (can approximate a probability accuracy up to 0.001%). We also consider the approximation method for small sized objects (ASO) proposed in [118] and described earlier. In ASO, the collision probability is evaluated by:

$$P_C\left(x_{i,k}, x_{j,k}\right) = V_s \frac{\exp\left[-\frac{1}{2}\left(\Delta x_{j,k} + \mathrm{m}_{i,k} - \mathrm{m}_{j,k}\right)^T \left(\Sigma_{i,k} + \Sigma_{j,k}\right)^{-1} \left(\Delta x_{j,k} + \mathrm{m}_{i,k} - \mathrm{m}_{j,k}\right)\right]}{\sqrt{(2\pi)^{n_x} \left|\Sigma_{i,k} + \Sigma_{j,k}\right|}} \quad (5.33)$$

where *Vs* is the volume of $\mathrm{X}_{ij,k}(0)$.

First, we start with a comparison of the closeness evaluation methods. When applying NI and MSC (with 100000 samples) in closeness evaluation, the same group structure evolution profile is obtained (see Figure 5-8). However, no group structure evolution is found in the ASO case (not plotted here); with ASO, the group index of all IOVs remain at 0 as individuals. These can be explained by the evaluated closeness profiles

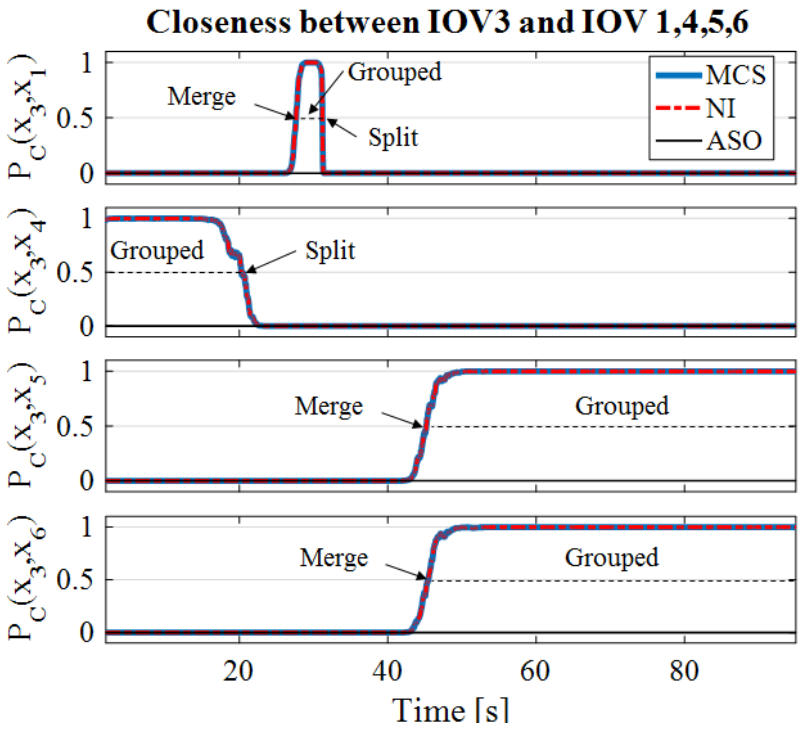


Figure 5-9: Closeness between IOV 3 and some of the other IOVs under the Monte Carlo Simulation(MSC) method with 100000 samples, numerical integration (NI) method, and the approximation method for small-sized object (ASO).

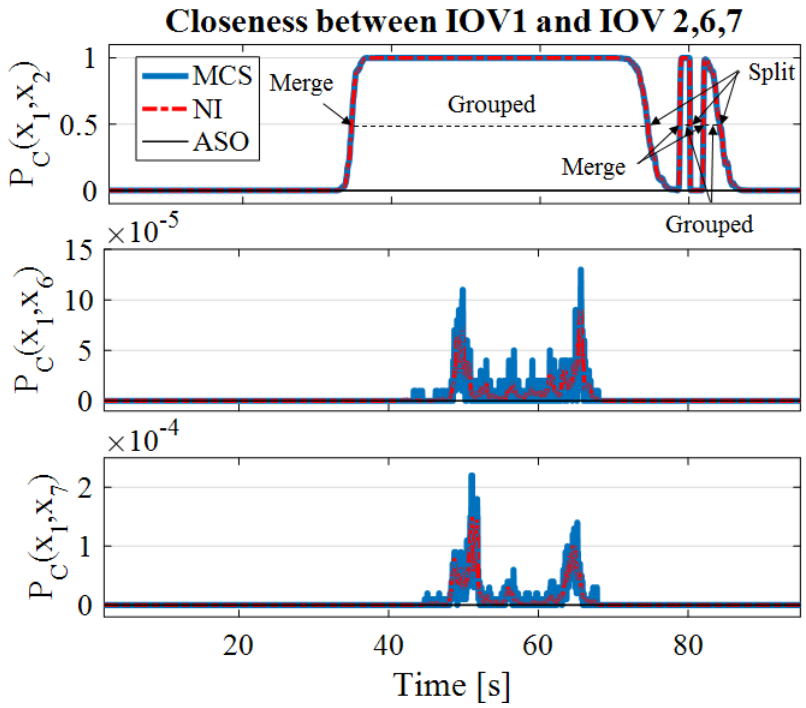


Figure 5-10: Closeness between IOV1 and some of other IOVs under the three methods

between each pair of IOVs under the three methods, as shown in Figure 5-9 and Figure 5-10, herein, the closeness evaluated by NI and MSC are almost the same (with accuracy up to 0.001%). In addition, the RMS of the errors between the closeness evaluation using MSC with different samples and NI, ASO and NI for the IOVs shown in Figure 5-9 and Figure 5-10 are given in Table 5-2. As more samples are used, the error between the closeness evaluation MSC and NI become smaller, which demonstrate the validity of our NI method. However, for ASO, the closeness is always 0 and the RMS error between NI and ASO is large. This is due to the fact that the size of the IOV is too large compared with the distribution area covered by the uncertainties. From (5.33), it's obvious that ASO uses the probability density value at the center of $\mathbb{X}_{ij,k}(0)$ to represent the density value for the

other locations in $\mathbb{X}_{ij,k}(0)$. This only works when the size of object, i.e. the volume of $\mathbb{X}_{ij,k}(0)$, is small. Otherwise, we obtain zero density value when the two objects are too far away relative to the distribution area arising from the uncertainties. Therefore, ASO is not suitable to use in cases with non-negligible geometric sizes of the objects involved (i. e, real highway vehicles).

The computational time for evaluating the closeness between a pair of IOVs under the three methods are also summarized in Table 5-3 (on a notebook PC with Intel i5-4200M 2.4 GHz processor and 4GB RAM). ASO is most efficient, but as described above is least accurate. The proposed NI gives a reasonably efficient resolution of the closeness evaluation when a high accuracy of probability evaluation, e.g. smaller than 0.01%, should be ensured. MSC is most accurate but is unlikely to be useable for real-time applications.

In Figure 5-8, we can see the total number of OVGs identified evolves as 2-1-2-1-2-3-2-3-2-3-2, as the states for the IOVs evolve differently. Initially, IOVs 1, 2, 5 and 6 are all SOVs as they are far away from other IOVs in positions. IOV 1 and IOV 2 are also different form other IOVs in velocity, see Figure 5-7. While IOVs 3, 4 are grouped in OVG 1 (see Figure 5-9 for the closeness values) and IOV 7, 8 are in OVG 2 due to their closeness both in positions and velocity.

As time goes, SOV1 changes lane to overtake IOV 3, 4 around t=10s and then IOV 3 starts to accelerate and move towards SOV 5 and SOV 6. As a result, OVG 1 splits into SOV 3 and SOV 4 around t= 20s, and OVG 2 becomes OVG 1. During the acceleration, SOV 3 is caught up by SOV 1 and they temporally merge into the new OVG 1 at t=27s. As IOV 1 tries to keep its fast speed, it decides to change lane to follow the faster IOV 2

Table 5-2: RMS of the error between other methods and NI in evaluating the closeness of two IOVs shown in Figure 5-9 and Figure 5-10.

RMS Error	**Method**			
	MCS (100000 samples)	**MCS (10000 samples)**	**MCS (1000 samples)**	**ASO**
IOV3 and IOV 1	0.00018	0.0005	0.0014	0.1848
IOV3 and IOV4	0.00033	0.0012	0.0034	0.4402
IOV3 and IOV5	0.00030	0.0010	0.0030	0.7173
IOV3 and IOV6	0.00027	0.0009	0.0030	0.7163
IOV1 and IOV2	0.00041	0.0013	0.0039	0.664
IOV1 and IOV6	5.52e-6	1.79e-5	6.01e-5	1.02e-5
IOV1 and IOV7	7.61e-6	2.30e-5	7.46e-5	1.75e-5

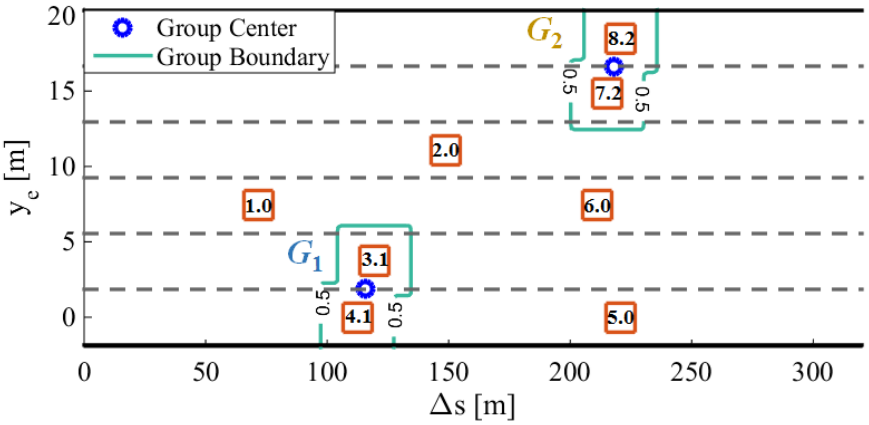
Table 5-3: Execution time of the error in evaluating the closeness of two IOVs for 3168 runs under different methods on a notebook with Intel i5-4200M 2.4 GHz processor and 4GB RAM.

Execution Time	**Method**				
	MCS (100000 samples)	**MCS (10000 samples)**	**MCS (1000 samples)**	**NI**	**ASO**
Max [ms]	33.9	1.16	0.232	0.762	0.036
Min [ms]	11.5	0.634	0.116	0.483	0.018
Mean [ms]	11.7	0.717	0.123	0.495	0.019

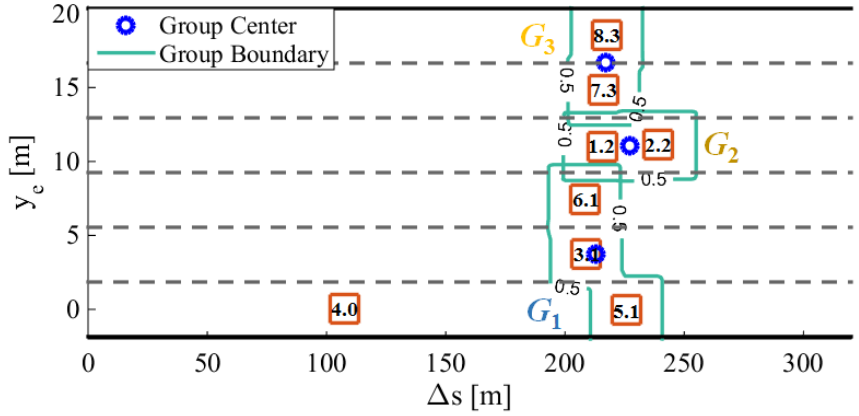
among the rest of the IOVs to pass through the traffic jam formed by IOVs 5, 6, 7 and 8 (The details of the motion control decisions are discussed in our other work [124]). Therefore, the OVG 1 split again at t=31s and SOV 1 and 2 merge into the new OVG 1 at t=33s, see Figure 5-10. Later SOV 3 decelerate to merge with SOV 5 and SOV 6 to form a new OVG 2 at t=45s. When a faster IOV passes by a slower IOV through an adjacent lane, we can see the rise of closeness between them, for example, the closeness of IOV 1 and 6 and IOV 1 and 7 rises when IOV1 is passing through the traffic jam as can be seen in Figure 5-10. However, as can be seen in Figure 5-7, the longitudinal velocity gap between them is too large (two times larger than the specified bound of the velocity difference Δv_t) to significantly increase the closeness.

After IOV1 pass through the traffic jam together with IOV 2 in OVG 2, it changes lane to overtake IOV 2 and then OVG 2 splits into SOV 1 and SOV 2 at t=74s. However, IOV 2 moves laterally towards the lane occupied by IOV 1. To avoid a collision, IOV 1 decelerates and change lane to the right. When IOV 2 settles down, IVO 1 accelerates to overtake it. During this collision avoidance process, we can see the oscillation of the closeness value between IOV 1 and IOV 2 in Figure 5-10 between t=74~85s. This is mainly due to the rapid velocity change of IOV 1 to avoid a collision and then overtake the slower IOV 2.

We can also generate the group geometry description for this scenario. Figure 5-11 shows an example for the position description sampled at time t=15 and 60s. Here, we use a belief contour with α=0.5 to represent the distribution of the OVGs. The OVG position center calculated by applying equations (5.25) is also shown. Each IOV is labeled with



(a) t=15s



(b) t=60s

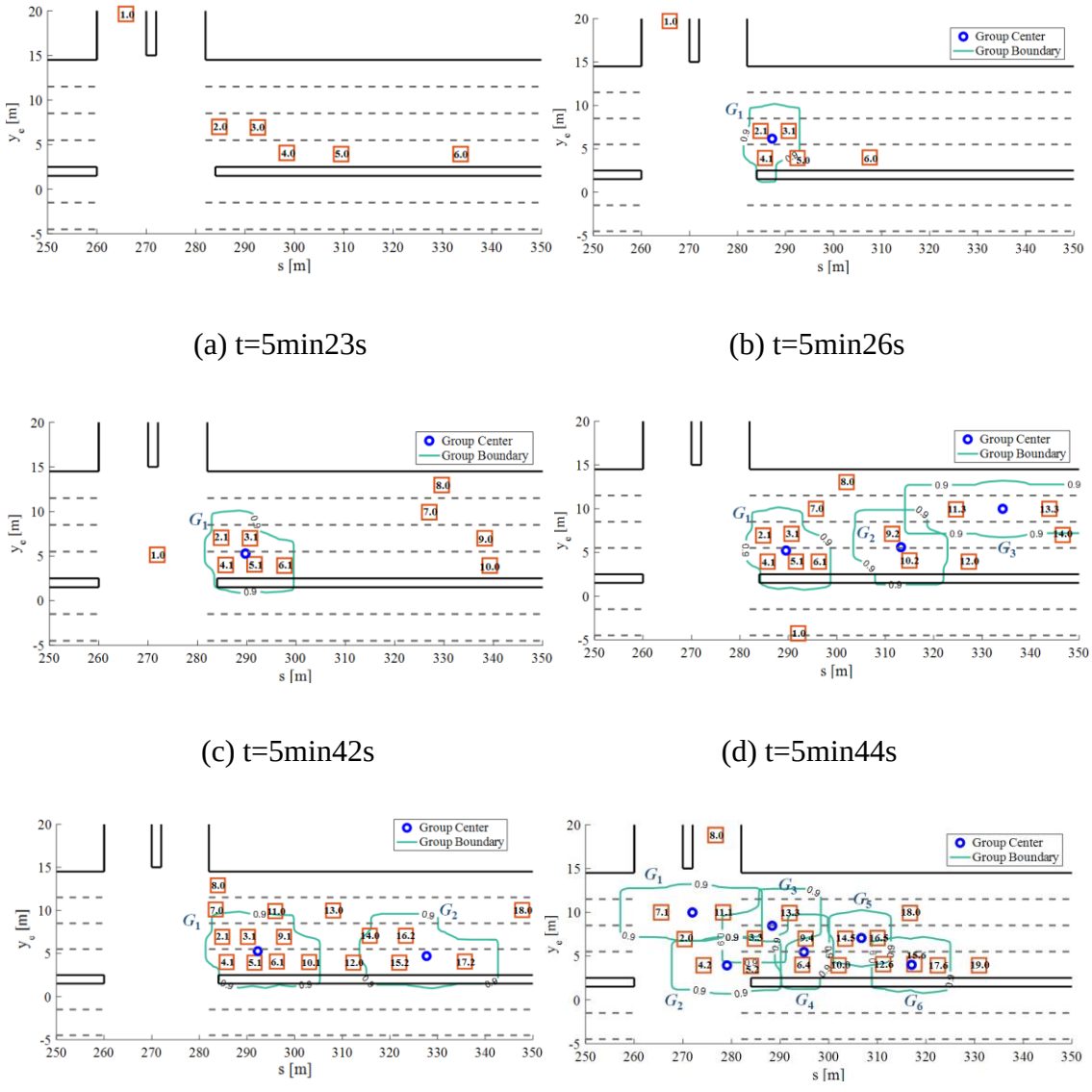
Figure 5-11: Example of relative position description of OVGs at different time for highway scenario. See Figure 5-2 for adopted numbering convention.

their group index behind their IOV index. We can see how the group boundary contour and center position changes along with the indices of the IOVs in the OVG from t=15 to 60s, as shown in Figure 5-11.

5.4.2 Application to Real-Time Traffic Intersection Data Set

For simplicity and to avoid duplication, in the intersection scenario, we only apply our grouping function with the NI method of closeness evaluation. We apply the grouping function to the vehicle trajectory data generated by the NG-VIDEO software after

processing a 15 min overhead camera record of actual traffic in an intersection on Lankershim Boulevard in Los Angeles, CA (data from [123]). Here, we chose one segment from t = 5min23s to 5min52 to show the typical vehicle grouping results as an example. The position description of the OVGs are sampled at 6 frames, as shown in Figure 5-12. We can clearly see the group formation (OVG1) of a set of IOVs (IOV2~IOV6) from SOVs at t = 5min23s to a unique group at t = 5 min 42 (Figure 5-12 (a)~(c)) when these IOVs stop before a red light. The closeness profile between the front and rear IOVs in this case are not given here, but it's similar to the case of IOV 3 and IOV5 or IOV6 in the highway scenario above with velocity synchronization (0 m/s) and position proximity at the end. Then, more and more IOVs slow down to approach OVG1 with different velocities from behind. Among these IOVs, the ones with similar velocities and close distance merge into groups, as shown in the group identities in Figure 5-12 (c)~(e). After IOV1 completes its left turn, the light turns green, and the set of IOVs start to pass the intersection with some distinct or some similar accelerations, therefore, big OVGs split into small OVGs, as shown in Figure 5-12 (f). We can see the grouping method successfully identifies the IOVs with common and distinct motion and accordingly adjusts the group identities in this intersection scenario.



(a) t=5min23s (b) t=5min26s

(c) t=5min42s (d) t=5min44s

(e) t=5min48s f) t=5min52s

Figure 5-12: Example of position description of OVGs at different time for the intersection scenario under NI method in closeness evaluation. See Figure 5-2 for the

5.4.3 Comments on the Application of Grouping to Motion Planning for Autonomous Vehicles

As the group boundary represents the probability of the distribution of the IOVs in the OVG, it can be used to re-define the multitude of obstacle avoidance constraints that arise in the real-time motion planning of autonomous vehicles in uncertain public traffic involving many vehicles. Such planning frameworks were discussed, for example, our previous works [31], [124] that discuss predictive control approaches and even those of [51]that use rapidly exploring random trees.

In this chapter, we considered probabilistic collision problems involving IOVs with non-negligible geometric shapes and derived the condition given in equation (5.16) and given a simplified derivation in *Proposition 1* for the case of Gaussian distribution of the motion states. If we only apply the chanced constraints derived from these results for collision avoidance of the Autonomously Controlled Vehicle (ACV or ego-vehicle) with each detected IOV, in the numerical optimization problem of predictive motion planning, some undesirable local minimums will result the intersections of the collision boundaries of IOVs (with some closeness) that will trap the ACV from finding better solutions for its motion plan. We have shown earlier that for deterministic planning [93] [116], a good OVG algorithm can help to tailor the feasible field to exclude these effects. In principle, one can expect this to work for the probabilistic planning case as well since the chanced constraints (representing avoidance of IOVs and OVGs) can be numerically transformed to the deterministic constraints for the optimization problem to find a solution [125].

However, the real-time motion planning problem requires a real-time solution of the group boundary generation, like the ASO method. To determine the group contours for the illustrations in this chapter, we used numerical methods (integrations or simplified solution mentioned in *Remark 1*) to sample a set of points based on the collision probability evaluation. While this represents the true boundary, its computation may not be efficient for real-time implementation in all scenarios. Therefore, in our continuing work, we approximate the probabilistic collision condition and the contour of the OVGs with conservative closed-form results, similar as the OVG boundary for the deterministic case in [93] [116], and apply it in stochastic motion planning algorithms for autonomous vehicles.

5.5 Conclusion

In this chapter, we propose a probabilistic multiple vehicle grouping framework for tracking the evolution of groups of individual object vehicles (IOVs) with the consideration of their non-negligible geometric sizes and prevalent sensing and motion uncertainties. Therein, the closeness between any two IOVs, which is defined by a probabilistic collision condition comprising of mutual proximity both in velocities and positions as the main criteria for subsequent clustering of detected vehicles into object vehicle groups (OVGs) whose states are estimated by the weighted distribution of each IOV in the OVG. The workings and performance of our proposed framework are illustrated for a simulated complex scenario and a real-time traffic intersection dataset. Comparison of the probabilistic collision condition as derived and evaluated via a numerical integration method with Monte Carlo Simulations show that it can achieve very good accuracy with

about a 20x computational speed up. It is also highlighted that while computationally more efficient approaches of closeness evaluation that ignore geometric sizes exist, they could not resolve group attributes and are not applicable for road vehicles.

In next chapter we will focuses on finding better approximations of the closeness evaluation to further reduce its computational complexity and on determining the OVG boundary so that it can be executed in real-time within stochastic motion planning algorithms for autonomous vehicles.

CHAPTER 6

PROBABILISTIC COLLISION AVOIDANCE FOR AN AUTONOMOUS VEHICLE IN MULTI-VEHICLE TRAFFIC

6.1 Abstract

Self-driving and highly automated vehicles are expected to reliably avoid collision with other traffic participants. However, due to uncertainties from sensor imperfections and environmental disturbances, collision avoidance conditions are often expressed as difficult to resolve probabilistic constraints in the motion planning problem. In this paper, we propose a constraint tightening method to progressively transform the probabilistic collision condition to a conservative hyper-elliptical condition that can be computed efficiently within real-time ready predictive motion planning algorithms. This is done via a conservative closed-form transformation of the bivariate integral in the collision probability density function and subsequent computable approximations with logistic functions. To further facilitate the motion planning these vehicles in multi-vehicle traffic, individual object vehicles with similar velocities and with proximity in position can be grouped via a clustering scheme that draws on these ideas. Detailed numerical experiments are included to illustrate the workings and the performance of the proposed approach when incorporated in stochastic model-predictive motion planning methods.

6.2 Introduction

In the march towards (semi-)autonomous driving, the task of safely guiding the controlled vehicle in the presence of other traffic participants remains a challenging

problem. Therein, using the sensors to track/observe the states, like position and velocities of the other objects plays a significant role, as it provides the basic but most important information for the autonomously controlled vehicle (ACV) to avoid a collision.

To identify individual object vehicle from sets of measurements, e.g., sparse laser point cloud, it's better to track each vehicle as an extended object with simple geometric shapes like circle [103], ellipse [104], rectangle [100]or such arbitrary shapes [107]. Data association approaches like Multi Hypothesis Tracking (MHT) [92], Probabilistic MHT (PMHT) [109], Probability Hypothesis Density (PHD) approach [110], Joint Probabilistic Data Association (JPDA) approach [111], or Random Finite Sets (RFS) [112] can be used to assign the measurements to each identified object vehicle. In addition, by considering the knowledge of the geometric vehicle model, i.e. fused with camera image, the detected object vehicle are finally represented by an estimated geometric/spatial shape (center and content parameters) and dynamics (location and velocity) [113] .

The geometric shapes mentioned above are usually used in the motion planning problem to formulate the collision avoidance constraint. However, this problem is challenging for the following three principle reasons: 1) the planning problem is non-convex as the feasible field is defined outside the area occupied by the object vehicles. 2) the planning problem naturally involves uncertainties due to model error, sensor imperfection or environmental disturbances. 3) the future information of the surrounding dynamic traffic is unknown, so is the uncertainty propagation. To address these challenges, several approaches are proposed.

For the first challenge, polygonal models [126] [94], as a disjunction of linear constraints, or algebraic models like circle, ellipses [95] and hyper ellipses [96] are mostly used in the sampling based planning method like RRT* algorithm [51]. However, for mathematical constrained optimization based planning method like MPC [31], algebraic models are better options than polygonal models because the disjunction of linear constraint will lead to discontinuity in the state space, which results in the Disjunctive Linear Programming problem [127]. This problem is similar as Mixed-Integer Programming problem that requires specific solver to find a solution and is not efficient for real time planning.

For the second challenge, the uncertainties can be handled by either considering their bounds (non-deterministic case) or distribution (probabilistic case) [128]. In the non-deterministic case [74], the worst case of the uncertainty is considered in the motion planning problem thus leading to a too conservative solution for the planning problem. However, in the probabilistic case [126], the conservatism can be flexibly adjusted by specifying a chanced constrain with a confidence threshold/coefficient, but the computations are often intractable due to the multivariate integrals. Nevertheless, for specified confidence threshold/coefficient (e.g. probability of collision less than some small value), a solution can be obtained by solving the approximate deterministic motion planning problem with either constraint tightening method [86] [129]or sampling-based method [126] that account for the uncertainties. By contrast, the sampling-based method involves prohibitive computational expense due to the large number of samples for the uncertainty and thus is not appropriate for real time application.

For the constraint tightening method, a deterministic constrain described by a function of the level of confidence and the distribution of the uncertainty is used to replace the original chanced constrain. Therefore the key process is to determine the function. With assumptions of uncorrelated Gaussian distributions for each state observation, an explicit function is easy to obtain due to the single-variate integral [86]. If the state uncertainties are correlated, an approximate explicit function for probabilistic collision evaluation was given in [118] for small sized objects (radius smaller than 1 m) on a plane surface. However, it doesn't work for a normal sized vehicle. Thus a numerical method was developed in [130] and then efficiently approximated via rotation strategy and logistic function in [129] to evaluate the probability of collision between two individual object vehicles (IOVs) with non-negligible geometric shapes/sizes. However, the resulting explicit function written in the combinations of logistic function is still too complex to be involved in the optimization based planning method.

For the third challenge, the future states of the surrounding traffic can be model-predicted for a certain prediction horizon based on the current measurements. This allows the ACV to assess the risk of having a collision with other IOVs and then to determine a collision-free trajectory. Different models used for motion prediction of OVs are summarized in [65], including physics-based models [66, 67] for short term prediction(less than 1 second), maneuver-based models [68, 69] and interaction-aware models [70, 71] for longer-term prediction. When state uncertainties are considered in the prediction horizon, several methods are proposed depending on the type of model and uncertainty used in the prediction. If the state uncertainties are Gaussian distributed, filtering techniques, e.g.,

Kalman filter (KF) series [78, 79] (regular KF for linear model or Uscented KF for nonlinear model) for motion prediction of one maneuver and Interactive Multi Model (IMM) KF or Switching KF [80] for different possible maneuvers can be applied. If non-Gaussian uncertainties are considered, approximation method such like particle sampling method [131] or polynomial chaos expansion [132] can be used to propagate the uncertainties in the prediction horizon.

For a certainty amount of IOVs that have common movement (e.g. similar velocities) and close geometrical proximity, they can be regarded as an object vehicle group (OVG) and represented by an extended moving object. This helps to redefine the feasible collision-free field to exclude undesired local minimums for the motion plan [93]. The OVG can be formed either based on a distance threshold defined by the overlap of their elliptical collision fields for a deterministic case [116] or on a probabilistic closeness threshold defined by a similarity function in terms of the position and velocity distribution for the case with state uncertainties [130] [129]. For the case with uncertainties, the boundary of the group is then defined as the joint state distribution of those IOVs in the group with a specified confidence coefficient. This can be used to set up a chanced constraint to avoid collision with an OVG. However, it's not yet implemented with any kind of motion planning method. Furthermore, a more efficient explicit function is required to approximate the tightened constraint of using the logistic functions, for the purpose of real-time implementation with optimization based motion planning methods.

The objective of this chapter is to find a more efficient way to tighten the probabilistic collision avoidance constraint and implement it with an existing predictive motion planning method. The contribution of this chapter includes:

- Develop a constraint tightening strategy via deriving the analytical solution for the fundamental dimension of the collision area according to the specified confidence threshold and the uncertainty distribution.
- Implement the proposed constraint tightening method to multi-vehicle grouping frame work and solve the stochastic MPC based motion planning problem for collision avoidance and traffic interaction.

6.3 Tightening the Constraint of Probabilistic Collision Avoidance

Let $\mathbb{X}(x)$ be the state space (position) occupied by a vehicle considering its geometric shape. Without too much loss of generality, hereafter, x represents only the position state for the centroid of the geometric shape of each vehicle, including ACV and IOV. Then, the collision between ACV and IOV i at time k (the subscript k is omitted in the following content for simplicity) is defined by the condition $C(x_A,x_{O\_i})$: $\mathbb{X}_A(x_A)\cap\mathbb{X}_{O\_i}(x_{O\_i})\neq\emptyset$, as is shown in Figure 5-3 with rectangular geometric shape description . Then the probabilistic for ACV to avoid a collision with IOV i is higher than a specified confidence value $1-\delta$, $0<\delta<1$, can be given by:

$$P\left(X_A(x_A)\cap X_{O\_i}\left(x_{O\_i}\right)\neq\emptyset\right)\leq\delta \tag{6.1}$$

To simplify the evaluation of collision probability we do some modification to the collision condition as well as the collision avoidance constraint as shown below:

$$P\left(x_A \in \mathbb{X}_{A,O\_i}\left(x_{O\_i}\right)\right) \le \delta \tag{6.2}$$

where $\mathbb{X}_{A,O\_i}(x_{O\_i})$ is an extended geometric space occupied by the IOV i at time k on which we lump the geometric shapes/sizes of ACV $\mathbb{X}_A(x_A)$ and IOV i $\mathbb{X}_{O\_i}(x_{O\_i})$. Therein, ACV is considered as a point. Note that (3) and (4) are equivalent. An example of collision in 2D position space between ACV and IOV i with rectangular shapes is shown in Figure 5-3. One can also similarly derive the extended shape $\mathbb{X}_{A,O\_i}(x_{O\_i})$ for other geometric descriptions like circles or ellipses. But the rectangular shapes can help to find the close-form solution for the probability of collision.

Assuming the Gaussian distribution on the position states, a general integral form to evaluate the probabilistic collision between the ACV and an IOV with non-negligible geometric shape (rectangular) is derived in the *Proposition 1* in Section 4.3.3. However, to avoid the complexity of using numerical evaluation of (5.17), here we adopt an explicit formula to approximate (5.17) for rapid evaluation and real-time application with MPC based motion planning framework.

Considering the collision in a 2D case, the position state is defined by:

$$x = \begin{bmatrix} s \\ y_e \end{bmatrix} \tag{6.3}$$

where s and y_e are the arch length and lateral position for the centroid of the vehicle geometric shape defined in the Frenet frame, as shown in Figure 2-4. By following the assumptions used in *Proposition 1* and rectangular shape for the vehicles, (5.17) can be rewritten as a definite bivariate normal integral:

$$P\left(x_A \in \mathbb{X}_{A,O\_i}\left(x_{O\_i}\right)\right) = P\left(\Delta\underline{s} < \Delta s_{O\_i} < \Delta\overline{s},\ \Delta\underline{y}_e < \Delta y_{e,O\_i} < \Delta\overline{y}_e\right) =$$
$$\int_{\Delta\underline{s}}^{\Delta\overline{s}}\int_{\Delta\underline{y}_e}^{\Delta\overline{y}_e} \frac{\exp\left[-\frac{1}{2}\left[\Delta s_{O\_i},\ \Delta y_{e,O\_i}\right]\Sigma^{-1}_{A,O\_i,k}\left[\Delta s_{O\_i},\ \Delta y_{e,O\_i}\right]^T\right]}{\sqrt{(2\pi)^{n_x}\left|\Sigma_{A,O\_i}\right|}} d\Delta s_{O\_i} d\Delta y_{e,O\_i} \tag{6.4}$$

where the integral interval ($\Delta\overline{s}$, $\Delta\underline{s}$), ($\Delta\overline{y}_e$, $\Delta\underline{y}_e$) are defined by:

$$\begin{aligned}
\Delta\overline{s} &= \mathrm{m}_{s,A} - \mathrm{m}_{s,O_i} + a_A + a_{O_i} \\
\Delta\underline{s} &= \mathrm{m}_{s,A} - \mathrm{m}_{s,O_i} - a_A - a_{O_i} \\
\Delta\overline{y}_e &= \mathrm{m}_{ye,A} - \mathrm{m}_{ye,O_i} + b_A + b_{O_i} \\
\Delta\underline{y}_e &= \mathrm{m}_{ye,A} - \mathrm{m}_{ye,O_i} - b_A - b_{O_i}
\end{aligned} \tag{6.5}$$

and the combined covariance matrix Σ_{A,O_i} is given by:

$$\Sigma_{A,O\_i} = \begin{bmatrix} \sigma^2_{s,A,O\_i} & \rho\sigma_{s,A,O\_i}\sigma_{ye,A,O\_i} \\ \rho\sigma_{ye,A,O\_i}\sigma_{s,A,O\_i} & \sigma^2_{ye,A,O\_i} \end{bmatrix}$$
$$\begin{aligned}
\sigma_{s,A,O_i} &= \sigma^2_{s,A} + \sigma^2_{s,O_i} \\
\sigma_{ye,A,O_i} &= \sigma^2_{ye,A} + \sigma^2_{ye,O_i}
\end{aligned} \tag{6.6}$$

where σ is the variance for each position state and ρ is the correlation coefficient between s and y_e. Without loss of generality, here we consider $|\rho|<1$. To efficiently and conservatively evaluate the probability of collision in (6.4) with $|\rho|<1$, we give the following closed-form approximation that voids the evaluation of the integral on the right-hand side of (6.4). We detail the derivations with general observations about bivariate normal distributions in *Remark* 6-1 and *Remark* 6-2.

Remark 6-1: Let (Δs, Δy_e) have a bivariate normal distribution with correlation coefficient $\rho=0$:

$$\begin{bmatrix} \Delta s \\ \Delta y_e \end{bmatrix} \sim N\left(\begin{bmatrix} m_{\Delta s} \\ m_{\Delta y_e} \end{bmatrix}, \begin{bmatrix} \sigma_s^2 & 0 \\ 0 & \sigma_{y_e}^2 \end{bmatrix} \right) \tag{6.7}$$

The solution for the bivariate integral within the integral region ($\Delta\overline{s}$, $\Delta\underline{s}$), ($\Delta\overline{y}_e$, $\Delta\underline{y}_e$) is easy to obtain:

$$\begin{aligned} &P\left(\Delta\underline{s} \le \Delta s \le \Delta\overline{s}, \Delta\underline{y}_e \le \Delta y_e \le \Delta\overline{y}_e\right) = \\ &\frac{1}{4}\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma_s}\left(\Delta s - m_s\right)\right)\Bigg|_{\Delta\underline{s}}^{\Delta\overline{s}} \operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma_{y_e}}\left(\Delta y_e - m_{y_e}\right)\right)\Bigg|_{\Delta\underline{y}_e}^{\Delta\overline{y}_e} \end{aligned} \tag{6.8}$$

where erf is the error function:

$$\operatorname{erf}\left(x\right) = \frac{1}{\sqrt{\pi}}\int_{-x}^{x} e^{-t^2}\,\mathrm{d}\,t \tag{6.9}$$

Remark 6-2*:* Let (*X*, *Y*) have a bivariate normal distribution with zero means and correlation coefficient $|\rho|<1$:

$$\begin{bmatrix} \Delta s \\ \Delta y_e \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_s^2 & \rho\sigma_s\sigma_{y_e} \\ \rho\sigma_{y_e}\sigma_s & \sigma_{y_e}^2 \end{bmatrix} \right) \tag{6.10}$$

It's hard to directly find the closed-form solution for the bivariate integral of this distribution, but it can be transformed into a bivariate normal distribution with ρ=0 (form of (6.7)) via a coordinate rotation, as shown in Figure 6-1. After rotation, the new distribution is given by:

$$\begin{bmatrix} \Delta s' \\ \Delta y_e' \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_{s'}^2 & 0 \\ 0 & \sigma_{y_e'}^2 \end{bmatrix} \right) \tag{6.11}$$

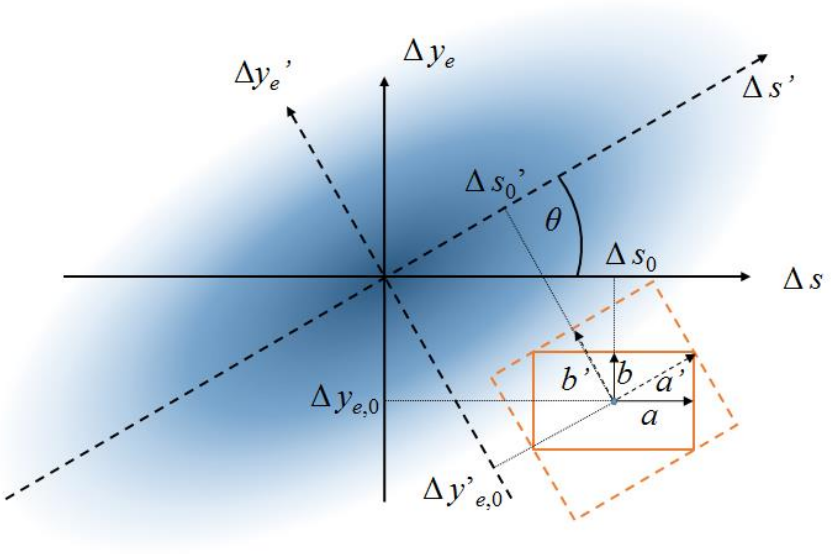


Figure 6-1: Bivariate normal distribution under different coordinate. $\rho \neq 0$ in $\Delta s/\Delta ye$ coordinate. $\rho=0$ in $\Delta s'/\Delta ye'$ coordinate. The red rectangle represents the integral region.

The new self-variances $\sigma_{s'}$, $\sigma_{ye'}$ and the rotational angle θ can be determined by computing the eigenvalues and eigenvector of the old covariance matrix [133] to arrive at:

$$\sigma_{s'}^2, \sigma_{y_e'}^2 = \frac{\sigma_s^2}{2} + \frac{\sigma_{y_e}^2}{2} \pm \frac{\sqrt{\sigma_s^4 + 4\rho^2\sigma_s^2\sigma_{y_e}^2 - 2\sigma_s^2\sigma_{y_e}^2 + \sigma_{y_e}^4}}{2} \tag{6.12}$$

$$\theta = -\arctan\left(\frac{\sigma_s^2}{2} + \frac{\sigma_{y_e}^2}{2} - \frac{\sqrt{\sigma_s^4 + 4\rho^2\sigma_s^2\sigma_{y_e}^2 - 2\sigma_s^2\sigma_{y_e}^2 + \sigma_{y_e}^4}}{2\rho\sigma_s\sigma_{y_e}} - \frac{\sigma_{y_e}}{\rho\sigma_s} \right) \tag{6.13}$$

Note that different from the case in *Remark* 6-1, the $\Delta s/\Delta y_e$ coordinate and $\Delta s'/\Delta y_e'$ coordinate are both coordinates with their origins located at the means of the bivariate distribution (or are cases with zero means; non-zero mean distributions can be easily handled by applying translations with the means).

To apply the solution of the bivariate integral obtained from *Remark* 6-1 to our problem, the rectangular integral region need to be rotated parallel with the axes of the new

coordinate system. Also, to ensure a conservative evaluation of the probability, the new integral region must cover or circumscribe the old one. As shown in Figure 6-1, considering the original integral region $[\Delta\underline{s}, \Delta\overline{s}]=[\Delta s_0-a, \Delta s_0+a]$, $[\Delta\underline{y}_e, \Delta\overline{y}_e]=[\Delta y_{e,0}-b, \Delta y_{e,0}+b]$ in $\Delta s/\Delta y_e$ coordinate, we select the new integral region to be given by $[\Delta\underline{s}', \Delta\overline{s}']=[\Delta s_0'-a', \Delta s_0'+a']$, $[\Delta\underline{y}_e', \Delta\overline{y}_e']=[\Delta y'_{e,0}-b', \Delta y'_{e,0}+b']$ in $\Delta s'/\Delta y_e'$ coordinate, where:

$$\begin{bmatrix} \Delta s_0' \\ \Delta y'_{e,0} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \Delta s_0 \\ \Delta y_{e,0} \end{bmatrix} \tag{6.14}$$

$$\begin{bmatrix} a' \\ b' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \tag{6.15}$$

Then, a closed-form approximation of the cumulative probability of a bivariate normal distribution defined as (6.4) is conservatively obtained by:

$$P\left(\Delta\underline{s} \le \Delta s \le \Delta\overline{s}, \Delta\underline{y}_e \le \Delta y_e \le \Delta\overline{y}_e\right) \le P\left(\Delta\underline{s}' \le \Delta s' \le \Delta\overline{s}', \Delta\underline{y}_e' \le \Delta y_e' \le \Delta\overline{y}_e'\right) = \frac{1}{4}\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma_{s'}}\Delta s'\right)\Bigg|_{\Delta\underline{s}'}^{\Delta\overline{s}'} \operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma_{y_e'}}\Delta y_e'\right)\Bigg|_{\Delta\underline{y}_e'}^{\Delta\overline{y}_e'} \tag{6.16}$$

Using (6.16), the probability of collision between ACV and IOV *i* in (6.4) can be easily evaluated using the conservative integral region in the rotated coordinate system.

When a confidence threshold δ is specified, a related collision area $\mathbb{X}_{A,O\_i,k,\delta}$ for ACV inscribed in a rectangle (see Figure 6-3) is determined as:

$$\mathbb{X}_{A,O\_i,\delta}\left(x_{O\_i}\right) = \left\{x_{A,k} \mid P\left(x_{A,k} \in \mathbb{X}_{A,O\_i}\right) \le \delta\right\} \tag{6.17}$$

In the collision area $\mathbb{X}_{A,O\_i,k,\delta}$, the dimension $\Delta X'$ and $\Delta Y'$ can be approximated by explicit functions in terms of the confidence threshold δ, the size of the aforementioned extended geometric area $\mathbb{X}_{A,O\_i,k}(x_{O\_i,k})$ occupied by IOV i and the distribution of the uncertainties. Therein, the logistic function with a residual error function $r(x)$ is used to approximate the error function:

$$\operatorname{erf}(x) \approx f_l(x) + r(x) = \frac{2}{1+e^{-c_l x}} - 1 + r(x) \tag{6.18}$$

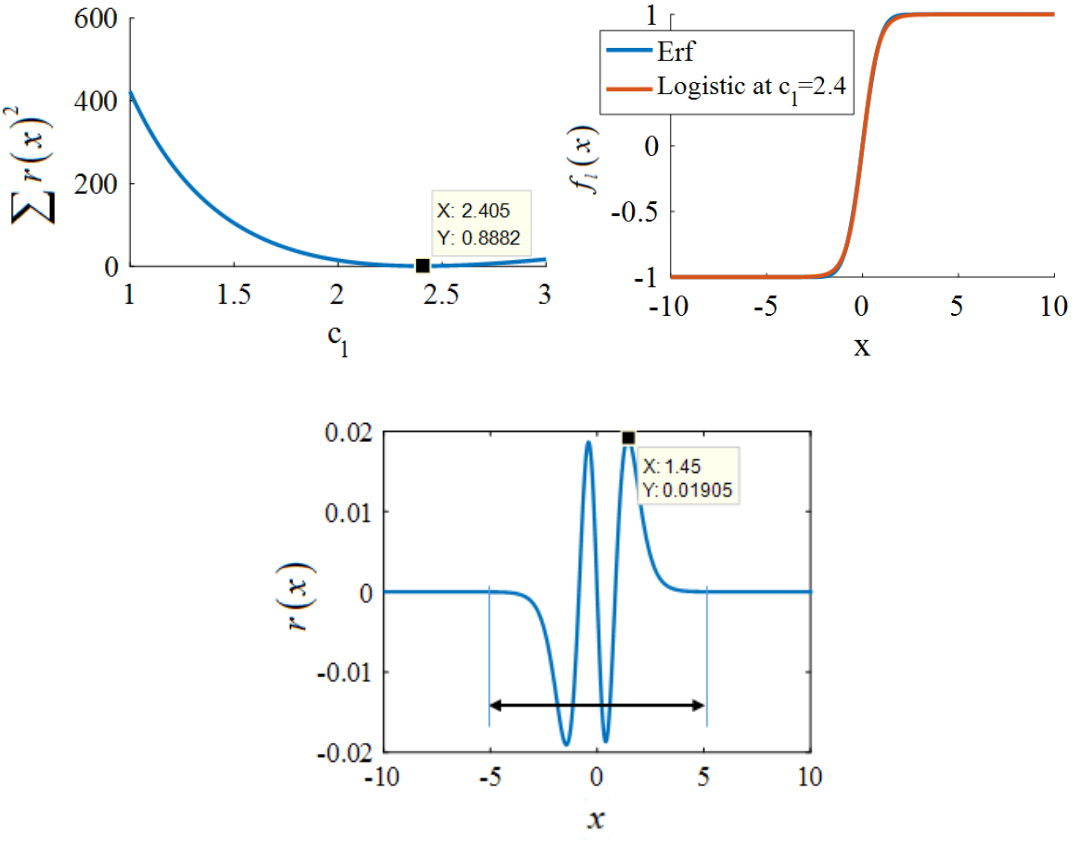


Figure 6-2: Approximation of the error function with a logistic function (Top left). The cumulative square error for 20000 samples range from -10 to 10 with different cl values (Top right). The approximation performance when cl=2.4. The maximum error is 0.019 when |x| is close to 1.45 (Bottom).

The minimum cumulative square error found with the coefficient c_I=2.4, as illustrated in Figure 6-2. $r(x)$ is symmetric with respect to the origin and only have non-zero values between the region [-5, 5]. Two monotonically decreasing regions including ($-\infty$,-1.45] and [1.45, $+\infty$) exist. The maximum bound for $r(x)$ is r_b = 0.019. These characteristics are useful in demonstrating the conservatism of the dimension for the collision area with uncertainties approximated by using the logistic function.

Remark 3: Let (Δs', Δy_e') have a bivariate normal distribution with correlation coefficient ρ=0 after rotation as described by (6.11), from (6.16), the boundary of the area (see Figure 6-3) with specified bivariate integral region $[\Delta\underline{s}', \Delta\overline{s}'] = [\Delta s_0'-a', \Delta s_0'+a']$, $[\Delta\underline{y}_e', \Delta\overline{y}_e'] = [\Delta y'_{e,0}-b', \Delta y'_{e,0}+b']$, $a', b' \geq 0$ and confidence threshold δ is defined by:

$$\frac{1}{4}\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma_{s'}}\Delta s'\right)\Bigg|_{\Delta\underline{s}'}^{\Delta\overline{s}'} \operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma_{y_e'}}\Delta y_e'\right)\Bigg|_{\Delta\underline{y}_e'}^{\Delta\overline{y}_e'} = \delta \tag{6.19}$$

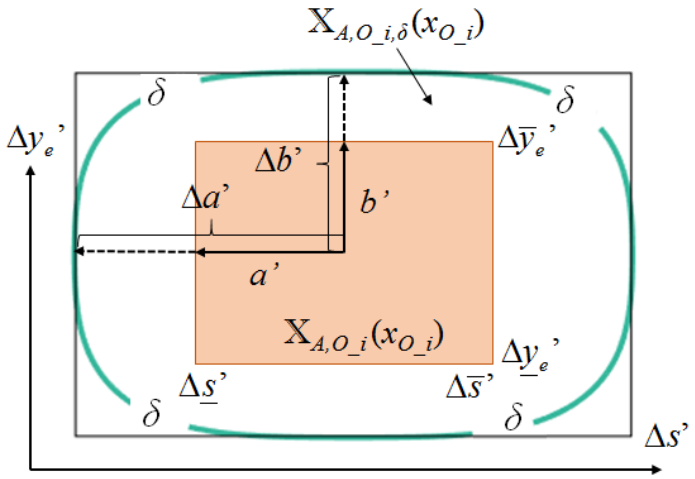


Figure 6-3: Collision area for ACV and IOV i with specified confidence threshold δ

From Figure 6-3 we can see that the shape of the area boundary is close to a hyper-ellipse inscribed in a rectangle with the same dimension for length Δa' and width Δb' as the major and minor length of the boundary. Also, from (6.19), the contribution of the distributions for Δs', Δy_e' to the cumulative probability are independent as they are decoupled, therefore calculating the length for Δa' and Δb' can be also decoupled, as given by:

$$\frac{C_{y_e}}{4}\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma_{s'}}\Delta s'\right)\Bigg|_{\Delta a'-a'}^{\Delta a'+a'}=\delta,\quad C_{y_e}=\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma_{y_e'}}\Delta y_e'\right)\Bigg|_{\Delta \underline{y}_e'}^{\Delta \bar{y}_e'} \tag{6.20}$$

$$\frac{C_s}{4}\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma_{y_e'}}\Delta y_e'\right)\Bigg|_{\Delta b'-b'}^{\Delta b'+b'}=\delta,\quad C_s=\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma_{s'}}\Delta s'\right)\Bigg|_{\Delta \underline{s}'}^{\Delta \bar{s}'} \tag{6.21}$$

As the error function is a single variate integral, Δa' and Δb' can't be analytically solved from (6.20) and (6.21). Therefore, we substitute the algebraic function (6.18) into (6.20) and (6.21) to replace the error function and obtain:

$$\frac{C_{y_e}}{4}\left[f_l\left(\frac{1}{\sqrt{2}\sigma_{s'}}\Delta s'\right)\Bigg|_{\Delta a'-a'}^{\Delta a'+a'}+r\left(\frac{1}{\sqrt{2}\sigma_{s'}}\Delta s'\right)\Bigg|_{\Delta a'-a'}^{\Delta a'+a'}\right]=\delta \tag{6.22}$$

$$\frac{C_s}{4}\left[f_l\left(\frac{1}{\sqrt{2}\sigma_{y_e'}}\Delta y_e'\right)\Bigg|_{\Delta b'-b'}^{\Delta b'+b'}+r\left(\frac{1}{\sqrt{2}\sigma_{y_e'}}\Delta y_e'\right)\Bigg|_{\Delta b'-b'}^{\Delta b'+b'}\right]=\delta \tag{6.23}$$

Before solving (6.22) and (6.23), the residual error function $r(x)$ must be handled. From Figure 6-2 and (6.16), we can see the two aforementioned monotonically decreasing regions $(-\infty,-1.45]$ and $[1.45,+\infty)$ are related to a small confidence threshold of collision.

Therefore, when δ is specified in a particular region close to zero, the residual error part in (6.22) and (6.23) will have negative value and thus result in a conservative solution for the dimension of the collision area. When δ is specified out of the region close to zero, the residual error will not always be negative. Therefore, a worst case compensation considering the maximum residual error bound r_b will be used to ensure the conservative solution. Above all, by considering the monotonicity, symmetry with respect to the origin and the bound of the function and the following scenario based equation is proposed to ensure the conservatism in determining the dimension of the confident collision area:

$$\text{if } \frac{2a'}{\sqrt{2}\sigma_{s'}} \geq 3.55: \begin{cases} \dfrac{C_{y_e}}{4} f_l\left(\dfrac{1}{\sqrt{2}\sigma_{s'}}\Delta s'\right)\Big|_{\Delta a'-a'}^{\Delta a'+a'} = \delta, \text{ if } 0<\delta \leq r_b \\ \dfrac{C_{y_e}}{4} f_l\left(\dfrac{1}{\sqrt{2}\sigma_{s'}}\Delta s'\right)\Big|_{\Delta a'-a'}^{\Delta a'+a'} = \delta - \dfrac{2r_b C_{y_e}}{4}, \text{ else} \end{cases}$$

$$\text{else if } \frac{2a'}{\sqrt{2}\sigma_{X'}} < 3.55: \begin{cases} \dfrac{C_{y_e}}{4} f_l\left(\dfrac{1}{\sqrt{2}\sigma_{s'}}\Delta s'\right)\Big|_{\Delta a'-a'}^{\Delta a'+a'} = \delta, \text{ if } 0<\delta \leq r\left(5 - \dfrac{2a'}{\sqrt{2}\sigma_{s'}}\right) \\ \dfrac{C_{y_e}}{4} f_l\left(\dfrac{1}{\sqrt{2}\sigma_{s'}}\Delta s'\right)\Big|_{\Delta a'-a'}^{\Delta a'+a'} = \max\left(\delta - \dfrac{2r_b C_{y_e}}{4}, \varepsilon\right), \text{ else} \end{cases} \quad (6.24)$$

$$\text{if } \frac{2b'}{\sqrt{2}\sigma_{y_e'}} \geq 3.55: \begin{cases} \dfrac{C_s}{4} f_l\left(\dfrac{1}{\sqrt{2}\sigma_{y_e'}}\Delta y_e'\right)\Big|_{\Delta b'-b'}^{\Delta b'+b'} = \delta, \text{ if } 0<\delta \leq r_b \\ \dfrac{C_s}{4} f_l\left(\dfrac{1}{\sqrt{2}\sigma_{y_e'}}\Delta y_e'\right)\Big|_{\Delta b'-b'}^{\Delta b'+b'} = \delta - \dfrac{2r_b C_s}{4}, \text{ else} \end{cases}$$

$$\text{else if } \frac{2b'}{\sqrt{2}\sigma_{y_e'}} < 3.55: \begin{cases} \dfrac{C_s}{4} f_l\left(\dfrac{1}{\sqrt{2}\sigma_{y_e'}}\Delta y_e'\right)\Big|_{\Delta b'-b'}^{\Delta b'+b'} = \delta, \text{ if } 0<\delta \leq r\left(5 - \dfrac{2b'}{\sqrt{2}\sigma_{y_e'}}\right) \\ \dfrac{C_s}{4} f_l\left(\dfrac{1}{\sqrt{2}\sigma_{y_e'}}\Delta y_e'\right)\Big|_{\Delta b'-b'}^{\Delta b'+b'} = \max\left(\delta - \dfrac{2r_b C_s}{4}, \varepsilon\right), \text{ else} \end{cases} \quad (6.25)$$

where $\varepsilon>0$ is a small value to ensure a feasible solution in analytically resolving the algebraic equation (6.24) and (6.25). When substituting the expression of logistic function in (6.24) and (6.25) and use a new variable to represent the exponential function for Δa' and Δb', two univariate quadratic equations are obtained and therefore easy to solve. For simplicity of the chapter, the final expressions are not given here.

Once Δa' and Δb' are solved, the rectangular collision area in Figure 6-3 for a specific confidence threshold δ is obtained and thus it can be in cooperated with the existing motion planning method like RRT*. However for mathematical constrained optimization based planning method like MPC, an algebraic model is better for the purpose of real-time implementation. Therefore, we bring the idea from [93] and [116] that using the fourth order hyper ellipse to describe the collision area between ACV and IOV i:

$$\frac{\left(\left(s_A - s_{O\_i}\right)\cos\theta - \left(y_{e,A} - y_{e,O\_i}\right)\sin\theta\right)^4}{\left(c_a \Delta a'\right)^4} + \frac{\left(\left(s_A - s_O\right)\sin\theta + \left(y_{e,A} - y_{e,O}\right)\cos\theta\right)^4}{\left(c_a \Delta b'\right)^4} = 1 \quad (6.26)$$

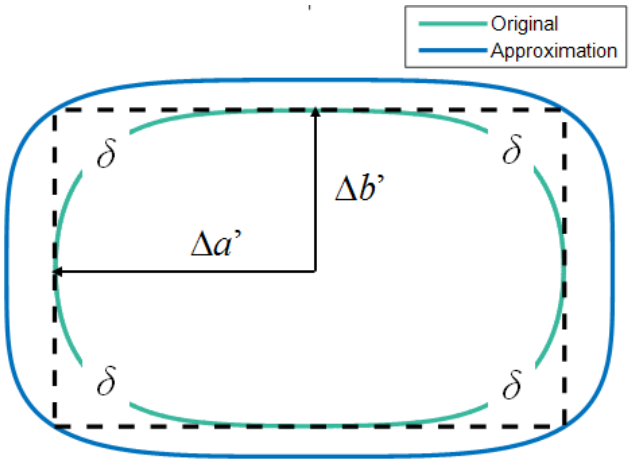


Figure 6-4: Algebraic collision area for ACV and IOV i described by a conservative fourth order hyper ellipse with c_a=1.189

Where c_a is a coefficient that can adjust the conservatism of the forth order hyper ellipse. With a determined rectangular (Δa' and Δb' for length and width and rotational angle θ), the circumscribed fourth order hyper ellipse with minimum area is related to c_a=1.189, which ensures a conservative collision area that includes the original collision area as a subset, as shown in Figure 6-4.

6.4 Multiple Vehicle Grouping Framework

In last chapter, we proposed a probabilistic framework to track groups of IOVs. From the probabilistic grouping results, the probability to avoid a collision with a certain OVG i can be described by the joint probability of collision avoidance with all the IOVs in the OVG i. Specifically, with a confidence value 1-δ, $0<\delta<1$, of ACV to avoid collision avoidance with OVG i at time k, the constraint is given by:

$$P\left(\bigcup_{j\in I_{G,i}} x_A \in \mathbb{X}_{A,O\_j}\left(x_{O\_j}\right) \right) \le \delta \tag{6.27}$$

According to the Boole's inequality, (6.27) can be conservatively converted to:

$$\sum_{j=1}^{N_{I_G,i}} P\left(x_A \in \mathbb{X}_{A,O\_j}\left(x_{O\_j}\right) \right) \le \delta \tag{6.28}$$

Where $N_{I_G,i}$ is the number of the IOVs in OVG i. We can see by following (6.28), evaluating the constraint for ACV to avoid a collision with OVG i could give rise to computational burden, when $N_{I_G,i}$ takes a large number, which might deteriorate the performance of real-time motion planning method like MPC or RRT* that requires high frequency for the update of the solution plan. To overcome this issue, here we consider the

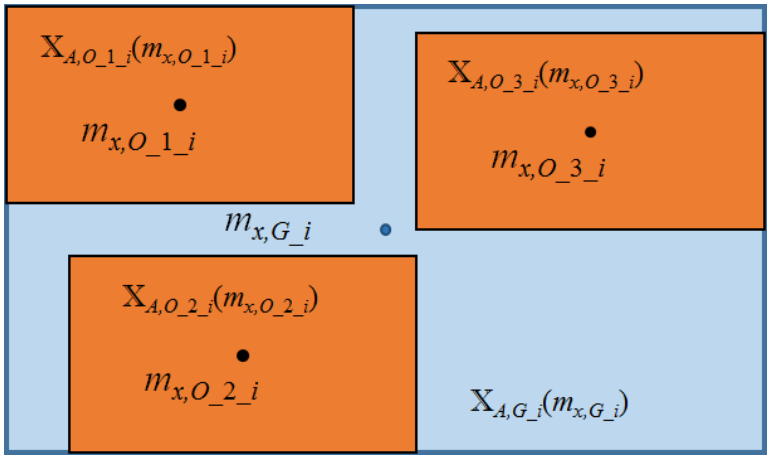


Figure 6-5: Illustration of the extended geometric shape for the OVG i

OVGs as an individual vehicle with extended shape. The idea also comes from our previous work [93] [116]. The extended shape $\mathbb{X}_{A,G\_i}(x_{G\_i})$ can be obtained from the union of the geometric shape of all the IOVs in the group. An example for rectangle shape description is shown in Figure 6-5. In this case, to avoid the collision with OVG *i* will be transformed to avoiding the collision with an extended IOV. Therefore (6.28) becomes:

$$P\left(x_A \in \mathbb{X}_{A,G\_i}\left(x_{G\_i}\right)\right) \le \delta \tag{6.29}$$

With Gaussian uncertainty assumption, (6.4)/(6.16) can be used to evaluate (6.29) with the integral intervals defined by:

$$\begin{aligned} \Delta\overline{s} &= \mathrm{m}_{s,G\_i} + a_{A,G\_i} \\ \Delta\underline{s} &= \mathrm{m}_{s,G\_i} - a_{A,G\_i} \\ \Delta\overline{y}_e &= \mathrm{m}_{ye,G\_i} + b_{A,G\_i} \\ \Delta\underline{y}_e &= \mathrm{m}_{ye,G\_i} - b_{A,G\_i} \end{aligned} \tag{6.30}$$

where

$$
\begin{aligned}
\mathrm{m}_{s,G\_i} &= \frac{\max\left(\mathrm{m}_{s,O\_I_{G,i}} + a_{A,O\_I_{G,i}}\right) + \min\left(\mathrm{m}_{s,O\_I_{G,i}} - a_{A,O\_I_{G,i}}\right)}{2} \\
\mathrm{m}_{ye,G\_i} &= \frac{\max\left(\mathrm{m}_{ye,O\_I_{G,i}} + b_{A,O\_I_{G,i}}\right) + \min\left(\mathrm{m}_{ye,O\_I_{G,i}} - b_{A,O\_I_{G,i}}\right)}{2} \\
a_{A,G\_i} &= \max\left(\mathrm{m}_{s,O\_I_{G,i}} + a_{A,O\_I_{G,i}}\right) - \mathrm{m}_{s,G\_i} \\
b_{A,G\_i} &= \max\left(\mathrm{m}_{ye,O\_I_{G,i}} + b_{A,O\_I_{G,i}}\right) - \mathrm{m}_{ye,G\_i}
\end{aligned}
\tag{6.31}
$$

From (6.30) and (6.31), the extended geomantic shape of OVG *i* object covers more area than the union of the IOVs in the group, it will sacrifice some feasible region. But it helps to exclude some local minimum and reduce the time to evaluate the collision constraint for the planning problem, which we will demonstrate via simulation in the next session.

According to (6.28), we can see the IOVs that are closer to the ACV in the OVG *i* will have more influences on the accumulative collision probability. In addition, accounting for the uncertainties on the position state of ACV, the combined covariance matrix Σ_{A,G_i} between ACV and OVG *i* can be selected from the set of the combined covariance matrix $\Sigma_{A,O\_I_{G,i}}$ between ACV and IOVs in OVG *i*, based on the Euclidean distance:

$$
\Sigma_{A,G\_i} = \Sigma_{A,O\_j_{MD\min,G,i}} \tag{6.32}
$$

where $j_{MD\min,G,i}$ is the index for the IOV in OVG *i* with minimum Euclidean distance to the ACV:

$$
MD_{\min,G,i} = \min_{j \in I_{G,i}} \left(\sqrt{\left(s_A - s_{O\_j}\right) + \left(y_{e,A} - y_{e,O\_j}\right)} \right) \tag{6.33}
$$

Therefore, by determining the size, estimated position and its covariance matrix for the extended geomantic shape of OVG *i*. (6.29) can be evaluated and transformed into a tightened deterministic constraint as we proposed in Section 5.3.

6.5 Simulation and Results

To illustrate the performance of the proposed constraint tightening method, it's incorporated with the MPC based motion planning framework proposed in [31] for static and dynamic obstacle avoidance scenarios with state observation uncertainties.

6.5.1 MPC Problem Formulation

As uncertainties are considered, the motion planning problem to be solved over the prediction horizon [0, H_p] results in a stochastic MPC problem, regarding a predictive trajectory guidance problem that following one maneuver (tracking one suit of reference lane and speed as stated in Chapter 2) is formulated by:

$$\min_{u} E\left(\sum_{k=1}^{N_p-1} \left\| y_k - r_k \right\|_P^2 + \sum_{k=0}^{N_p-1} \left\| u_k \right\|_R^2 + \left\| y_{N_p} - r_{N_p} \right\|_{P_t}^2 \right) \tag{6.34}$$

$$\text{subject to :} \quad \dot{x}_A = f\left(x_A, u, w\right),\ x \in X\ ,\ u \in U,\ w \in W \tag{6.35}$$

$$x_A(0) = x_{A,0} \tag{6.36}$$

$$y = Ax_A \tag{6.37}$$

$$\Pr\left(c_1\left(x_A, u\right) \geq 0\right) \geq \delta \tag{6.38}$$

$$c_2\left(x_A, u\right) \geq 0 \tag{6.39}$$

Here, the cost function minimizes the expectation of state tracking error and control efforts. x covers all the state variables of the ACV particle motion model which includes the positon states given by (6.3), the whole model can be found in [31]. X represents the state-space for x. x_0 denotes the current/initial state (measured and estimated). The estimation of the system outputs, namely the speed v_t and lateral position y_e of the ACV are grouped in vector y to check their reference r. P, P_t and R are the weighting matrices for the cost of the on-going and terminal tracking error, and the cost of control efforts, respectively. In (3.17), U denotes the admissible set for input u, which includes the input to ACV motion model and selection variables. W is the state space for noise/disturbance w. All the probabilistic collision avoidance constraint for single IOVs (6.2) or OVGs (6.29) are included in the probabilistic constraint to satisfy a belief coefficient δ, denoted by (3.20). These constraints will be tightened into deterministic constrain by the method we proposed in the previous sections for tractable solutions. Other deterministic inequality constraint like control limits are included in (3.21). As for a predictive maneuver planning problem, the MPC formulation will follow the definition given in (3.16)~(3.21).

The MPC problem above is solved by ACADO Toolkit with an active-set method-based solver. To efficiently find the solution of MPC, the warm-start strategy is widely used that applying the solution of the MPC problem at last step as the initial guess for the current step. However, for a nonconvex problem, the warm start strategy will easily guide the optimization into local minimum. For the motion planning problem with multiple IOVs, the collision fields for different IOV with position proximity can create a space with local

minimum when the object of ACV is moving forward. However, grouping the vehicles with closeness can help to exclude those local minimums.

6.5.2 Uncertainty Propagation

Given the nonlinear system model in (3.17), we adopt Unscented Kalman Filter (UKF) [85] to estimate the motion states of ACV in the presence of process and measurement uncertainty/noise. A regular KF can be used for state estimation of IOV for one maneuver (tracking a specific lane and speed). To account for other possible maneuvers of the OVs, the Interactive Multi-Model KF algorithm [80, 86] can be applied for OV state estimation. Here, we assume the filtering parameters are well captured from the driving data for the drivers of all OVs of all maneuvers.

Given the current estimates of the ACV and OV states, one can predict the evolution of the mean and covariance of the states for the whole length of the prediction horizon of the MPC. Here, we propagate uncertainty in the predicted states (for both ACV and OV) using the filtering techniques (UKF/KF) based on (3.17) with the notion of the most likely measurement. It is shown in [78] that the most likely measurement will not introduce bias in the system, thus it is useful to constrain the uncertainty propagation. Finally, note that the future inputs used in the motion prediction of the ACV will be taken from the previous planning results of the MPC.

6.5.3 Simulation Results

In the simulation, two different methods of handling the observation uncertainties are compared. In the first case labeled “W/O CT”, the expectation of the state will be considered as deterministic and constrain tightening will not be applied, therefore the

extended geometric area $\mathbb{X}_{A,O\_i,k}(x_{O\_i,k})$ will be directly used to generate the algebraic model in (6.26) with c_a=1.189, which is without constrain tightening. In the "W/ CT" cases, the stochastic MPC problem formulated by (3.16) and solved via the proposed constraint tightening in (6.26) with c_a =1.189 is considered.

5.5.3.1 Avoiding Single Static IOV

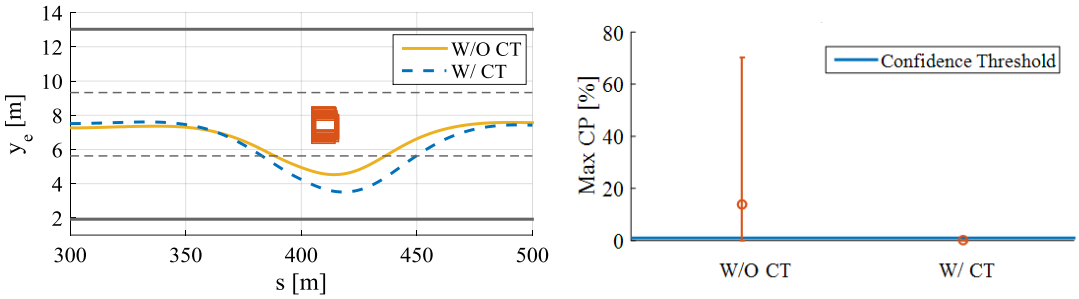
In this case, the ACV (at speed of 30m/s) needs to avoid a collision with a static IOV located at the same lane in front. Three different values of confidence threshold limit (δ=1%, 10% and 30%) are used to compare the performance of different uncertainty handling methods. As uncertainties are involved, for each δ setting, the simulation for each method will repeat 100 times to obtain the trajectories of the collision probabilities evaluated by (6.4) during the collision avoidance process. Then the maximum collision probabilities (Max CP) for the 100 trajectories will be recorded to fit a normal distribution model and then estimate the 99.9% confident interval of the MCP distribution. The main parameters for the recorded Max CP is summarized in Table 6-1. It should be noticed that, the different δ settings will not affect the distribution of the maximum collision probability

Table 6-1: Main parameters for the recorded Max CPs for different constraint handling method with different δ settings, obtained from 100 example simulations for each case

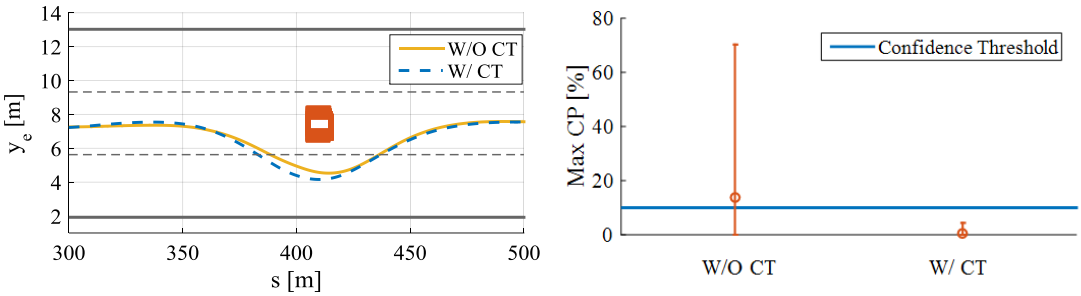
Max CP	**δ=1%**		**δ=10%**		**δ=30%**	
	Mean	**Max**	**Mean**	**Max**	**Mean**	**Max**
W/O CT	13.75%	70.2%	13.75%	70.2%	13.75%	70.2%
W/ CT	0.017%	0.26%	0.603%	4.38%	3.204%	20.41%

for W/O CT case as no uncertainties are considered. Therefore, the same distribution of Max CP of W/O CT case is used to compare with "W/ CT" cases for different δ settings.

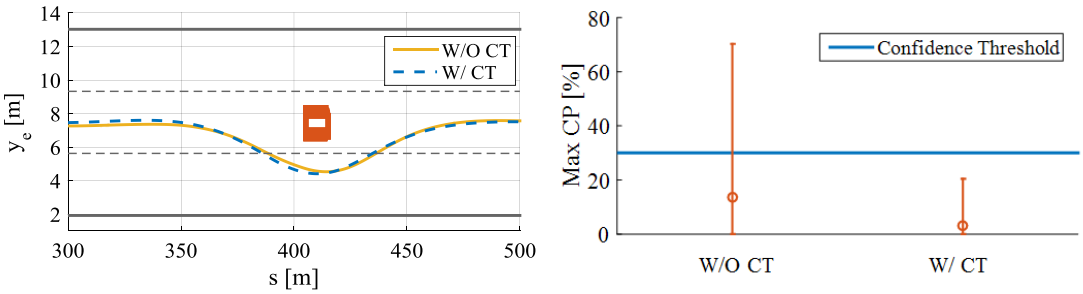
Figure 6-6 shows the trajectory examples and the Max CP distribution for different constraint handling methods at δ=1%, 10% and 30%. With higher Max CP, the position trajectory will stay closer to the IOV. We can see that as the W/O CT case has the fixed Max CP distribution. The confidence threshold will always be violated when δ is set to be less than some certain value (like around 30% in this simulation case), where the safety cannot be ensured. For the W/ CT cases, as the confidence threshold δ increases, the mean of the Max CP distribution rises and the 99.9% intervals extends. Also the position trajectories goes closer to the IOV. In the case W/ CT, the extension of the 99.9% interval is bounded and the confidence thresholds are always obeyed. This illustrate the conservatism of the collision area shown in Figure 6-4. The interval extension is mainly affected by the estimated characteristics of the uncertainties. Therefore, in the case with constraint handling, c_a =1.189 can ensure safety in the trajectory planning with a specified confident threshold.



a) δ=1%



b) δ=10%



c) δ=30%

Figure 6-6: Position trajectory examples (left) and the Max CP distribution (right, 99.9% confident range for normal distribution) for ACV to avoid a static IOV under different constrain handling methods

5.5.3.2 Avoiding Multiple Static IOVs

In this case, the ACV (at speed of 30m/s) needs to avoid three static IOVs with position proximity that create a local minimum for the MPC planner, as shown in Figure 6-7. The performace of planner with and without applying vehicle grouping method introduced in Section 5.3 as well as with different constraint handling methods are compared under δ=1%. The mean Max CP of all the recorded 100 examples for each case are summerized in Table 6-2.

Figure 6-7 shows the postion trajectory examples for the MPC planner without vehicle grouping method. We can see all the cases fail in overtaking the three IOVs due to the non-convex state space generated by the combination of the collison fields for each

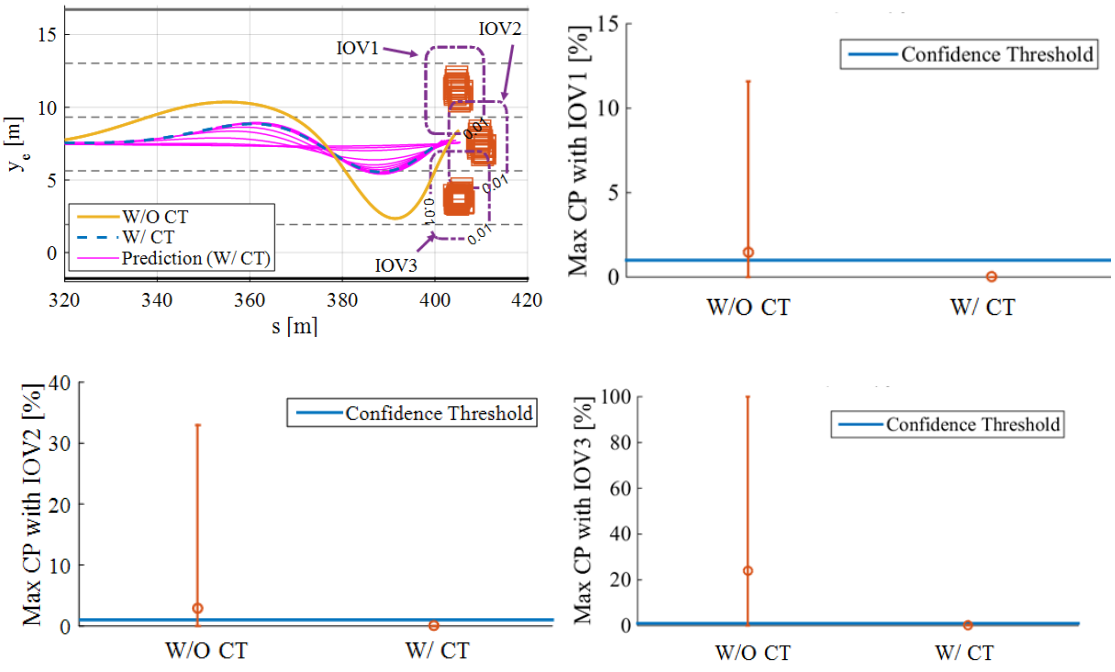


Figure 6-7: Position trajectory examples (top) and the Max CP probability distribution (bottom) for ACV to avoid multiple-IOV under different collision constraint settings without vehicle grouping (δ=1%)

Table 6-2: Mean of Max CPs for different constraint handling method in avoiding collision with multiple IOVs with δ=1%, obtained from 100 example simulations for each case

Mean Max CP	W/O Grouping			W/ Grouping			
	IOV1	IOV2	IOV3	IOV1	IOV2	IOV3	OVG
W/O CT	24.04%	2.93%	1.48%	0%	0%	0.489%	0.623%
W/ CT	$1.5e^{-4}$%	0%	0.05%	0%	0%	$4.4e^{-5}$%	$5.3e^{-5}$%

IOV(see the area inside the purple dash and dot envelop with specified confidence threshold 0.01 or 1%). It can be undestood via looking at the predicted trajectories for W/CT case. Initially the plan was going straight until the final position goes in to the collision field of IOV2. However, there is no lower cost direction to force the MPC planner with warm start strategy to jump out of the local minimum. Therefore, the planned final position will always stays in the local minimum step by step, and the ACV will be guided to slow down by following a curved path to balance the minimzation of the longitudinal and lateral deceleration and finally to stop at the area of local minimum. Due to different

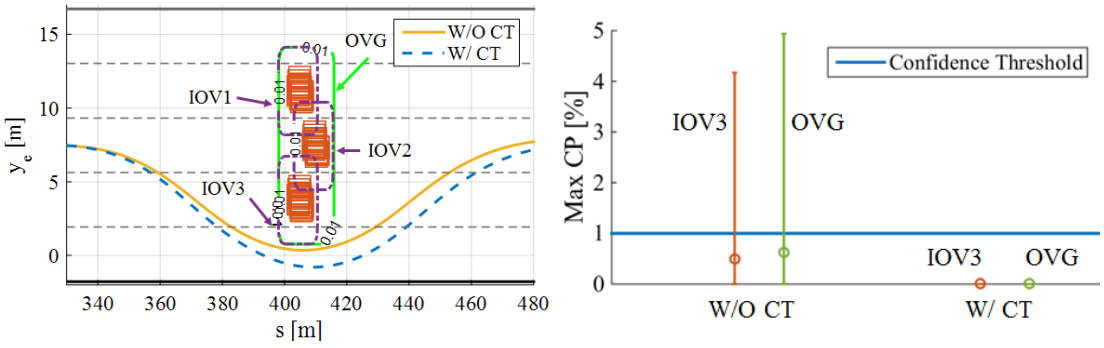


Figure 6-8: Position trajectory examples (top) and the Max CP probability distribution (bottom) for ACV to avoid multiple-IOVs under different collision constraint settings with vehicle grouping (δ=1%)

size of the collision field under differentconstraint handling methods, the resulted size of the area related to the local minima are different, which affects the shape of the planned trajectories. In the most conservative case of W/ CT, the planned trajectory is with the minimum curvature as the area for the loca minimum has the minimum size. This also means hardest brake is required in the deceleration for W/ CT comparing to W/O CT as its braking distance is the shorter. During the braking, only the Max CP with the IOVs for W/O CT case violate the confidence threshold, as shown in Figure 6-7 and Table 6-2.

While in Figure 6-8, as the vehicle grouping method is applied, a conservative hyper elliptical gourp boundary for the OVG is genereated to include all the collison fileds of each IOV and therefore excluding the undesired local minimum. All the cases with different constraint handlling mehods are able to overtake the multiple static vehicles without hard brake.The trajectoris of different cosntraint handling methods in this case are similar as the case of avoding a single static IOV as the vehicle group is consder as an extended single IOV. This also leads to a lower Max CP for avoiding IOV3 than avoiding the OVG, as the OVG has larger collision area, as shown in Table 6-2 and Figure 6-8 (right). In addition, we can see the Max CP in this case is smaller than the single IOV case. This is due to the fact that, the inscribed rectangle with bigger size will lead to more conservatism for the subscribed hyper elliptical collision area as a fixed conservatism coefficient c_a is used in (6.26).

5.5.3.3 Interaction with Multiple Dynamic IOVs

In this case, the ACV interacts with multiple IOVs with vehicle group changes due to lane change or speed change. As the performance of constraint tightening and vehicle

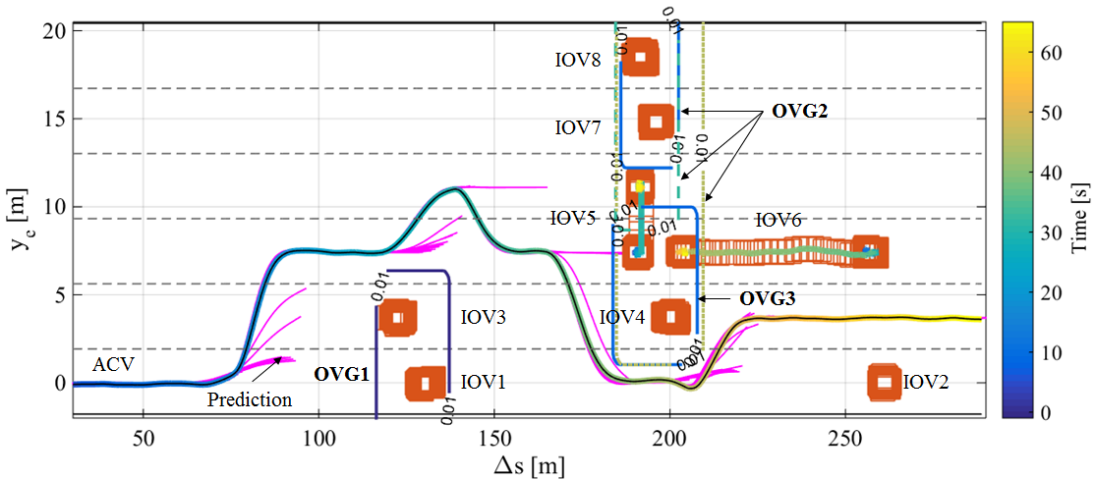
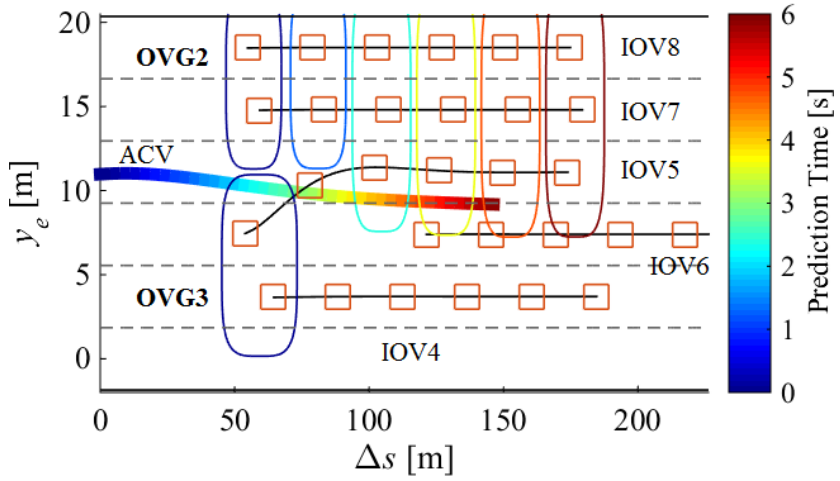


Figure 6-9: Position trajectory example for ACV to avoid multiple-IOVs guided by a MPC motion planner with collision constraint tightening (δ=1%) and vehicle grouping in highway scenario.

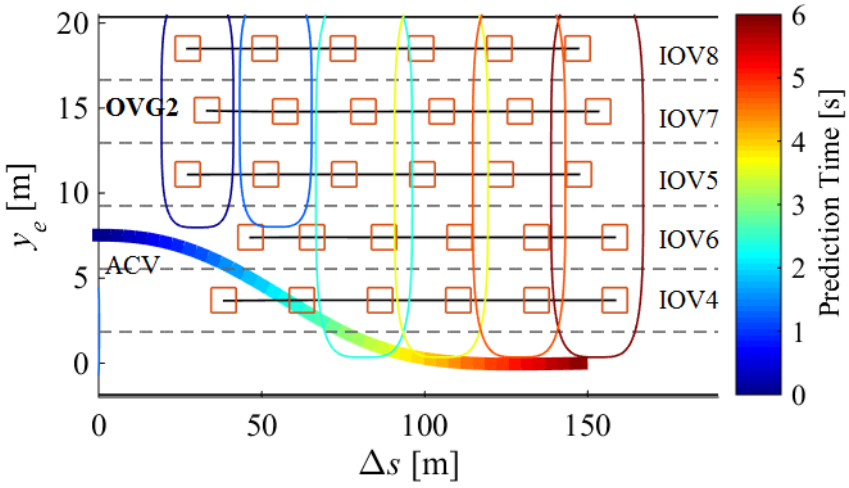
grouping on the motion planning of ACV have been illustrated in the previous collision avoidance cases, only the setting of W/ CT and W/ Grouping is use in this highway scenario. Here the MPC based motion planner proposed in [124]with ability of lane selection is used. A general scenario and the position trajectories for the ACV and IOVs is shown in Figure 6-9, where we can see how the ACV overtake 8 slower IOVs with fixed and variable OVGs in the front.

Initially, the ACV(at speed of 30m/s) makes a lane change to overtake the fixed OVG1 formed by IOV1 and IOV3(both at speed of 25m/s). Then it tries to make another lane change to the left adjacent one to overtake the original OVG3 formed by IOV4 and IOV5 (both at speed of 25m/s). But at t=30s, IOV5 changes lane to the left and OVG3 vanishes. When IOV5 arrives at the new lane, it merges with the original OVG2 formed by IOV7 and IOV8 and then generates the new OVG2, this can be seen in the trajectories of the prediction horizon shown in Figure 6-10a. These OVG changes affects the motion

plan of ACV that it decides to change back to the right lane to overtake the new OVG2. After that, since IOV6 slows down and then accelerate to fills the gap between the OVG2 (formed by IOV5, IOV7 and IOV8) and IOV4, the OVG2 is extended to include IOV4 and IOV6 inside. Therefore, the ACV plan to change lane to the most right one to overtake the extended OVG2, as shown in Figure 6-10b. Finally, the ACV changes lane to the left to



a) t=30.55s



b) t=35.88s

Figure 6-10: Examples of predicted trajectory for IOVs and planned trajectory for ACV with OVG boundary changes (δ=1%) in the predictive horizon

overtake IOV2 (at speed of 25m/s). From this scenario we can see how the OVG behaviors, including merge or vanish affects the motion planning of the ACV. During this process, the Max CP between ACV and the IOVs and OVGs are at very low values (close to zero) due to the conservative collision constraint tightening method applied.

6.6 Conclusion and Future Work

In this chapter we first develop a constraint tightening strategy that uses the logistic function and rotation approach to conservatively approximate the analytical solution for the fundamental dimension of the geometric collision area according to the specified confidence threshold and the uncertainty distribution. We then implement the proposed constraint tightening method to multi-vehicle grouping frame work and solve a stochastic MPC based motion planning problem for collision avoidance and traffic interaction. In the numerical experiment, we compare the approaches with and without using the constraint tightening method as well as vehicle grouping method. The statistic results shows that the specified confidence threshold for collision avoidance can be always satisfied when applying the proposed constraint tightening method. In addition, the vehicle grouping approach can help to exclude the undesired local minimum for the non-convex MPC based motion planning problem with a warm start strategy based numerical solver. Therefore, it can facilitate the motion planning of autonomous vehicle. In the future, we will work on finding a way to adjust the conservatism of the constraint tightening approach while also to ensure a safety.

CONCLUSIONS AND FUTURE WORK

1. Conclusion

The dissertation focus on predictive motion planning and control for autonomous driving with the consideration of 1) selection and planning of multiple maneuvers, 2) handling the stochastic uncertainties from state observation and 3) accommodating the dynamic environment with multiple object vehicles.

To deal with the above considerations, firstly, a hybrid predictive trajectory guidance framework is proposed that models the maneuvers of ego vehicle (tracking a specific reference speed on a specific lane) with particle description under predictive control as a hybrid control system. The multiple maneuver options provide the autonomous vehicle with flexibility and robustness in accommodating the complex traffic scenarios with proper behaviors comparing to the case with only one maneuver options such like collision avoidance.

Then the predictive reference speed assignment is combined with each optional reference lane in the configuration of the optimization of maneuver selection problem, thus realizing the predictive maneuver planning. Relaxation method is then applied to transform the mixed integer programming problem into nonlinear programming problem for real time application. Comparing with the hybrid predictive trajectory guidance framework, the predictive maneuver planning have better solutions in maneuver selection as the predictive motion of the surrounding traffic are considered and the switches among the optional maneuvers at each step of predictive horizon are optimized. However, this framework suffers from the local minimum do to the non-convex problem formulation. Therefore, a

forced maneuver selection (e.g. lane change) due to the reference speed adjustment is used to help the optimization problem jump out of the undesired local minimum. In addition, the obstacle filtering algorithm with vehicle grouping strategy is also used to effectively exclude the undesired local minimum via generating a more conservative collision field for the group of object vehicles with close proximity in positions.

When stochastic uncertainties are induced, a probabilistic evolution model is applied to track the structure, state and boundary of the object vehicle group. The probabilistic collision/closeness criteria is derived between any two individual object vehicles with non-negligible geometric size and motion uncertainties based on their state estimation via Bayesian tracking. Based on the closeness evaluation, a density-based method is applied to group/cluster the IOVs without a prior guess about the number of groups. The Monte Carlo Simulation with large samples (1000~100000) is used to demonstrate the correctness of the derived integral expression in evaluating the probabilistic closeness. In addition, while computationally more efficient approaches of closeness evaluation that ignore geometric sizes exist, they could not resolve group attributes and are not applicable for road vehicles.

Based on the probabilistic closeness/collision evaluation, a constraint tightening strategy is developed via deriving the approximated analytical solution for the fundamental dimension of the collision area with a specified confidence threshold and the estimated uncertainty distribution. The constraint tightening method is also extended to multi-vehicle grouping framework and implemented in solving the predictive maneuver planning problem with probabilistic collision avoidance constraint. The performance of the

constraint tightening strategy is demonstrated via statistic study based on the results of 100 Monte Carlo simulation samples, which shows that the specified confidence threshold can be always satisfied. Also, the vehicle group behaviors like merge and split clearly indicates the driving intention change of the object vehicles and therefore affects the maneuver plan of the autonomous vehicle.

2. Future work

In the future, the following problems need further exploration:

- The feasibility issue of the predictive maneuver planning with observation uncertainties. As the proposed predictive maneuver planning and control framework is based on the formulation of an MPC problem with various constraints. The successful implementation of the work is highly depends on the feasible solution found by the solver. However, in reality, due to the complexity of the dynamic environment (e.g. unexpected object vehicle maneuvers) and the uncertain condition of the sensing device (sensing range/view limitations, delays, faults and clutter), the feasible solutions might not be always found. Therefore, a feasibility study and perhaps some reactive approached to avoid an inevitable collision condition are still open for research.
- The accurate and efficient motion/interaction prediction of the detected object vehicles and vehicle groups. As the predictive maneuver/motion planning method require the future information of the surrounding traffic environment, the accurate, efficient prediction of the intention, motion of the other traffic participants are very important. Furthermore, the maneuver of ego vehicle will

have influence on the other object vehicles (ego vehicles or human driven vehicles), thus the interaction between them needs to be considered and predicted to improve the performance of the maneuver/motion planning framework.

- Conservatism of the constraint tightening approach. While with applying the constraint tightening approach, the specified confidence threshold is satisfied, the distribution of the collision probability are too conservative comparing to the confidence threshold, especially when the size of the geometric shape of the object vehicle is large(e.g. a vehicle group case). Therefore, a safe way to tune the conservatism of the constraint tightening approach is open for research.

APPENDIX

Sufficient Condition for Non-Overlapping of Two Ellipses with Parallel Axles

Use the parametric equation to describe the position of the ellipse defined in equation (4.3), (4.4), we obtain:

$$\begin{cases} x_{E1} = x_1 + a_1 \cos\theta_1 \\ y_{E1} = y_1 + b_1 \sin\theta_1 \end{cases} \tag{A.1}$$

$$\begin{cases} x_{E2} = x_2 + a_2 \cos\theta_2 \\ y_{E2} = y_2 + b_2 \sin\theta_2 \end{cases} \tag{A.2}$$

Considering the externally tangential condition of the two ellipses defined in equation (A.1),(A.2), as shown in Figure A-1, the position of the intersection between the two ellipses at the comment tangent should satisfied:

$$\frac{-b_1}{a_1}\cot\theta_1 = \frac{-b_2}{a_2}\cot\theta_2 \tag{A.3}$$

Assume ellipse 1 is fixed, combining equation (A.1)-(A.2), the algebraic equation for the center of ellipse 2 that externally tangential to the ellipse 1 can be derived by:

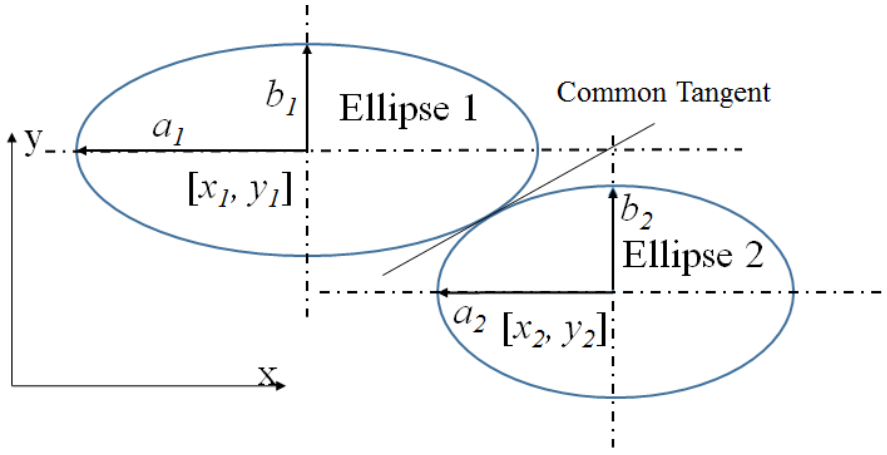


Figure A-1: External tangency of the two ellipses

$$\begin{cases} x_2 = x_1 + \left(a_1 + \dfrac{a_2^{\,2} b_1}{\sqrt{a_2^{\,2} b_1^{\,2} \cos^2 \theta_1 + a_1^{\,2} b_2^{\,2} \sin^2 \theta_1}} \right) \cos \theta_1 \\ y_2 = y_1 + \left(b_1 + \dfrac{a_1 b_2^{\,2}}{\sqrt{a_2^{\,2} b_1^{\,2} \cos^2 \theta_1 + a_1^{\,2} b_2^{\,2} \sin^2 \theta_1}} \right) \sin \theta_1 \end{cases} \tag{1}$$

which is bounded by

$$\begin{cases} \bar{x}_2 = x_1 + \left(a_1 + \dfrac{a_2^{\,2} b_1}{\min[a_1 b_2, a_2 b_1]} \right) \cos \theta_1 \\ \bar{y}_2 = y_1 + \left(b_1 + \dfrac{a_1 b_2^{\,2}}{\min[a_1 b_2, a_2 b_1]} \right) \sin \theta_1 \end{cases} \tag{2}$$

Thus the sufficient condition for the split of ellipse 2 from ellipse 1 can be defined by:

$$\left(\frac{x_2 - x_1}{a_1 + \dfrac{a_2^{\,2} b_1}{\min[a_1 b_2, a_2 b_1]}} \right)^2 + \left(\frac{y_2 - y_1}{b_1 + \dfrac{a_1 b_2^{\,2}}{\min[a_1 b_2, a_2 b_1]}} \right)^2 \geq 1 \tag{3}$$

Similarly, if ellipse 2 is fixed, the sufficient condition for the split of ellipse 1 from ellipse 2 can be defined by:

$$\left(\frac{x_1 - x_2}{a_2 + \dfrac{a_1^{\,2} b_2}{\min[a_1 b_2, a_2 b_1]}} \right)^2 + \left(\frac{y_1 - y_2}{b_2 + \dfrac{a_2 b_1^{\,2}}{\min[a_1 b_2, a_2 b_1]}} \right)^2 \geq 1 \tag{4}$$

Therefore, simultaneously satisfying equation (A.6) and (A.7) guarantees the non-overlap of the two ellipses with parallel axles.

REFERENCES

[1] A. Forrest and M. Konca, "Autonomous cars and society," Worcester Polytechnic Institute, 2007.

[2] A. Hars, " Autonomous cars: The next revolution looms," Inventivio Innovation Briefs, 2010.

[3] J. Anderson, "Autonomous Vehicle Technology: A Guide for Policymakers," RAND Corperation, 2014.

[4] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," DTIC Document, 1989.

[5] U. Muller, J. Ben, E. Cosatto, B. Flepp and Y. L. Cun, "Off-road obstacle avoidance through," in *Advances in neural information processing systems*, 2005.

[6] M. Bojarski. etc., "End to end learning for self-driving," in *arXiv preprint arXiv:1604.07316*, 2016.

[7] J. Zhang and K. Cho, "Query-Efficient Imitation Learning for End-to-End Autonomous Driving," in *arXiv preprint arXiv:1605.06450*, 2016.

[8] J. Greenhalgh and M. Mirmehdi, "Real-time detection and recognition of road traffic signs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 498-506, 2012.

[9] M. Aly, "Real time detection of lane markers in urban streets," in *IEEE Intelligent Vehicles Symposium*, 2008.

[10] C. Chen, A. Seff, A. Kornhauser and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[11] J. Beck and S. Christoph, "Non-parametric lane estimation in urban environments," in *Intelligent Vehicles Symposium IEEE Proceedings*, 2014.

[12] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez-Moreno and F. López-Ferreras, "Road-sign detection and recognition based on support vector machines," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 8, no. 2, pp. 264-278, 2007.

[13] C. Wang, "Simultaneous localization, mapping and moving object tracking," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, 2004.

[14] R. Daily and D. Bevly, "The use of GPS for vehicle stability control systems," *IEEE Transactions on Industrial Electronics,* vol. 51, no. 2, pp. 270-277, 2004.

[15] S. RJ, G. P, G. IS and T. N, "Robust algorithm for real-time route planning," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 36, no. 3, pp. 869-878, 2000.

[16] Q. Li, Z. Zeng and B. Yang, "Hierarchical model of road network for route planning in vehicle navigation systems," *IEEE Intelligent Transportation Systems Magazine,* vol. 1, no. 2, pp. 20-24, 2009.

[17] S. K. a. E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research,*, vol. 30, no. 7, p. 846–894, 2011.

[18] E. Frazzoli, M. A. Dahleh and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries," *Robotics, IEEE Transactions on,* vol. 21, no. 6, p. 1077–1091, 2005.

[19] R. G. Sanfelice and E. Frazzoli, "A hybrid control framework for robust maneuver-based motion planning," in *American Control Conference*, 2008.

[20] A. Liniger and J. Lygeros, "A viability approach for fast recursive feasible finite horizon path planning of autonomous RC cars," in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, Seattle, Washington, USA, 2015.

[21] E. Frazzoli, M. Dahleh and E. Feron, "Real-time motion planning for agile autonomous vehicles," in *American Control Conference*, 2001.

[22] J. Karimi and S. H. Pourtakdoust, "Optimal maneuver-based motion planning over terrain and threats using a dynamic hybrid PSO algorithm," *Aerospace Science and Technology,* vol. 26, no. 1, pp. 60-71, 2013.

[23] D. J. Grymin, C. B. Neas and M. Farhood, "A hierarchical approach for primitive-based motion planning and control of autonomous vehicles," *Robotics and Autonomous Systems,* vol. 62, no. 2, pp. 214-228, 2014.

[24] D. Ferguson, M. Likhachev and A. Stentz, "A guide to heuristic-based path planning," in *Proceedings of the international workshop on planning under uncertainty for autonomous systems, international conference on automated planning and scheduling (ICAPS)*, 2005.

[25] J. Maciejowski, Predictive Control: With Constraints, Harlow, UK: Prentice-Hall, 2002.

[26] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari and D. Hrovat, "MPC-based approach to active steering for autonomous vehicle systems," *International Journal of Vehicle Autonomous Systems,* vol. 3, no. 2-4, pp. 265-291, 2005.

[27] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *Control Systems Technology,* vol. 15, no. 3, pp. 566-580, 2007.

[28] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari and D. Hrovat, "A hierarchical model predictive control framework for autonomous ground vehicles," in *American Control Conference*, Seattle, Washington, USA, 2008.

[29] Y. Gao, T. Lin, F. Borrelli, E. Tseng and D. Hrovat, "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads," in *ASME 2010 Dynamic Systems and Control Conference*, Cambridge, Massachusetts, 2010.

[30] T. Weiskircher and B. Ayalew, "Frameworks for Interfacing Trajectory Tracking with Predictive Trajectory Planning for Autonomous Road Vehicle Control," in *American Control Conference*, Chicago, IL, USA, 2015.

[31] T. Weiskircher, Q. Wang and B. Ayalew, "A Predictive Guidance and Control Framework for (Semi-)Autonomous Vehicles in Public Traffic," *IEEE Transactions on Control Systems Technology*, vol. PP, no. 99, pp. 1-13, 2017.

[32] B. Houska, H. J. Ferreau and M. Diehl, "An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range," *Automatica*, vol. 47, no. 10, pp. 2279-2285, 2011.

[33] M. Vukov, A. Domahidi, H. J. Ferreau, M. Morari and M. Diehl, "Auto-generated algorithms for nonlinear model predictive control on long and on short horizons," in *IEEE 52nd Annual Conference on Decision and Control* , Firenze, Italy, 2013.

[34] A. R. and J. How, "Mixed-integer programming for control," in *American Control Conference*, 2005.

[35] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke and A. Mahajan, "Mixed-integer nonlinear optimization," *Acta Numerica*, vol. 22, pp. 1-131, 2013.

[36] V. L. Bageshwar, W. L. Garrard and R. Rajamani, "Model predictive control of transitional maneuvers for adaptive cruise control vehicles," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 5, pp. 1573-1585, 2004.

[37] A. Richards and J. How, "Mixed-integer programming for control," in *American Control Conference*, 2005.

[38] A. Kurt and Ü. Özgüner, "Hierarchical finite state machines for autonomous mobile systems," *Control Engineering Practice*, vol. 21, no. 2, pp. 184-194, 2013.

[39] V. Gadepally, A. Kurt, A. Krishnamurthy and Ü. Özgüner, "Driver/vehicle state estimation and detection," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, 2011.

[40] A. Girault, "A hybrid controller for autonomous vehicles driving on automated highways," *Transportation Research Part C: Emerging Technologies*, vol. 12, no. 6, p. 421–452, 2004.

[41] J. Lygeros, D. N. Godbole and S. Sastry, "Verified hybrid controllers for automated vehicles," *IEEE transactions on automatic control*, vol. 43, no. 4, pp. 522-539, 1998.

[42] R. Verma and D. D. Vecchio, "Safety control of hidden mode hybrid systems," *Automatic Control, IEEE Transactions on*, vol. 57, no. 1, pp. 62-77, 2012.

[43] T. Weiskircher and B. Ayalew, "Predictive Control for Autonomous Driving in Dynamic Public Traffic," in *American Control Conference*, Chicago, IL, 2015.

[44] H. J. Ferreau, Model Predictive Control Algorithms for Applications with Millisecond Timescales, KU Leuven: Doctoral dissertation, 2011.

[45] R. Quirynen, M. Vukov, M. Zanon and M. Diehl, "Autogenerating microsecond solvers for nonlinear mpc: A tutorial using acado integrators," *Optimal Control Applications and Methods*, vol. 36, no. 6, p. 685–704, 2015.

[46] D. Delling, P. Sanders, D. Schultes and D. Wagner, "Engineering route planning algorithms," in *Algorithmics of Large and Complex Networks*, vol. 5515, Springer Berlin Heidelberg, 2009, pp. 117-139.

[47] D. González, J. Pérez, V. Milanés and F. Nashashibi, "A Review of Motion Planning Techniques for Automated Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135-1145, 2016.

[48] B. Paden, M. Cap, S. Z. Yong, D. Yershov and E. Frazzoli, "A Survey of motion planning and control techniques for self-driving urban vehicles," arXiv preprint arXiv:1604.07446 , 2016.

[49] S. M. LaValle, Planning Algorithms, Cambridge University Press, 2006.

[50] A. Kushleyev and M. Likhachev, "Time-bounded lattice for efficient planning in dynamic environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[51] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105-1118, 2009.

[52] J. Bohren, T. Foote, J. Keller, A. Kushleyev, D. Lee, A. Stewart, P. Vernaza, J. Derenick, J. Spletzer and B. Satterfield, "Little ben: The ben franklin racing team's entry in the 2007 DARPA urban challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 598-614, 2008.

[53] D. Ferguson, T. M. Howard and M. Likhachev, "Motion planning in urban environments," *Journal of Field Robotics*, vol. 25, no. 11-12, pp. 939-960, 2008.

[54] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to continuous-curvature paths," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 1025-1035, 2004.

[55] M. McNaughton, C. Urmson, J. M. Dolan and J.-W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[56] L. Han, H. Yashiro, H. T. N. Nejad, Q. H. Do and S. Mita, "Bezier curve based path planning for autonomous vehicle in urban environment," in *IEEE Intelligent Vehicles Symposium*, 2010.

[57] G. Prokop, "Modeling human vehicle driving by model predictive online optimization," *Vehicle System Dynamics*, vol. 35, no. 1, pp. 19-53, 2001.

[58] J. Ziegler, P. Bender, T. Dang and C. Stiller, "Trajectory planning for Bertha—A local, continuous method," in *IEEE Intelligent Vehicles Symposium* , 2014.

[59] S. J. Anderson, S. C. Peters, T. E. Pilutti and K. Iagnemma, "An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios," *International Journal of Vehicle Autonomous Systems*, vol. 8, no. 2-4, pp. 190-216, 2010.

[60] A. Bar Hillel, D. L. Ronen Lerner and G. Raz, "Recent progress in road and lane detection: a survey," *Machine vision and applications,* vol. 25, no. 3, p. 727–745, 2014.

[61] T. Kowsari, S. S. Beauchemin and M. A. Bauer, "Map-based lane and obstacle-free area detection," in *IEEE International Conference on Computer Vision Theory and Applications (VISAPP)*, 2014.

[62] A. Gray, Y. Gao, T. Lin, J. K. Hedrick, H. E. Tseng and F. Borrelli, "Predictive control for agile semi-autonomous ground vehicles using motion primitives," in *American Control Conference*, Montréal, Canada, 2012.

[63] D. Q. Mayne, J. B. Rawlings, C. V. Rao and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica,* vol. 36, no. 6, pp. 789-814, 2000.

[64] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," in *IEEE 1997 European Control Conference (ECC)*, 1997.

[65] S. Lefèvre, D. Vasquez and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *Robomech Journal,* vol. 1, no. 1, 2014.

[66] R. Schubert, E. Richter and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *IEEE International Conference on Information Fusion*, 2008.

[67] M. Brannstrom, E. Coelingh and J. Sjoberg, "Model-based threat assessment for avoiding arbitrary vehicle collisions," *IEEE Transactions on Intelligent Transportation Systems,* vol. 11, no. 3, pp. 658-669, 2010.

[68] S. Atev, G. Miller and N. P. Papanikolopoulos, "Clustering of vehicle trajectories," *IEEE Transactions on Intelligent Transportation Systems,* vol. 11, no. 3, pp. 647-657, 2010.

[69] Q. Tran and J. Firl, "Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression," in *IEEE Intelligent Vehicles Symposium Proceedings*, 2014.

[70] A. Lawitzky, D. Althoff, C. F. Passenberg, G. Tanzmeister, D. Wollherr and M. Buss, "Interactive scene prediction for automotive applications," in *IEEE Intelligent Vehicles Symposium*, 2013.

[71] G. Agamennoni, J. I. Nieto and E. M. Nebot, "A Bayesian approach for driving behavior inference," in *IEEE Intelligent Vehicles Symposium* , 2011.

[72] N. Dadkhah and B. Mettler, "Survey of motion planning literature in the presence of uncertainty: Considerations for UAV guidance," *Journal of Intelligent & Robotic Systems,* vol. 65, no. 1-4, pp. 233-246, 2012.

[73] C. Schmidt, F. Oechsle and W. Branz, "Research on trajectory planning in emergency situations with multiple objects," in *IEEE Intelligent Transportation Systems Conference*, 2006.

[74] Y. Kuwata, T. Schouwenaars, A. Richards and J. How, "Robust constrained receding horizon control for trajectory planning," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005.

[75] P. Falcone, M. Ali and J. Sjoberg, "Predictive threat assessment via reachability analysis and set invariance theory," *IEEE Transactions on Intelligent Transportation Systems,* vol. 12, no. 4, pp. 1352-1361, 2011.

[76] M. Althoff, O. Stursberg and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems,* vol. 10, no. 2, pp. 299-310, 2009.

[77] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics,* vol. 30, no. 4, pp. 903-918, 2014.

[78] N. Du Toit and J. W. Burdick, "Robot motion planning in dynamic, uncertain environments," *IEEE Transactions on Robotics,* vol. 28, no. 1, pp. 101-115, 2012.

[79] M. Farrokhsiar and H. Najjaran, "Unscented model predictive control of chance constrained nonlinear systems," *Advanced Robotics,* vol. 28, no. 4, pp. 257-267, 2014.

[80] E. Mazor, A. Averbuch, Y. Bar-Shalom and J. Dayan, "Interacting multiple model methods in target tracking: a survey," *IEEE transactions on aerospace and electronic systems,* vol. 34, no. 1, pp. 103-123, 1998.

[81] M. S. Branicky, V. S. Borkar and S. K. Mitter, "A unified framework for hybrid control: Model and optimal control theory," *Automatic Control, IEEE Transactions on,* vol. 43, no. 1, pp. 31-45, 1998.

[82] M. Tsogas, A. Polychronopoulos, N. Floudas and A. Amditis, "Situation refinement for vehicle maneuver identification and driver's intention prediction," in *IEEE International Conference on Information Fusion*, 2007.

[83] V. Gadepally, A. Krishnamurthy and U. Ozguner, "A framework for estimating driver decisions near intersections," *IEEE Transactions on Intelligent Transportation Systems,* vol. 15, no. 2, pp. 637-646, 2014.

[84] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011.

[85] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI* , Orlando, FL, USA, 1997.

[86] A. Carvalho, Y. Gao, S. Lefevre and F. Borrelli, "Stochastic predictive control of autonomous vehicles in uncertain environments," in *International Symposium on Advanced Vehicle Control*, 2014.

[87] T. Streubel and K. H. Hoffmann, "Prediction of driver intended path at intersections," in *IEEE Intelligent Vehicles Symposium*, 2014.

[88] Q. Wang, B. Ayalew and T. Weiskircher, "Optimal Assigner Decisions in Predictive Control of an Autonomous Road Vehicle," in *American Control Conference*, Boston, MA, USA, 2016.

[89] Q. Wang, T. Weiskircher and B. Ayalew, "A Hierarchical Hybrid Predictive Control of an Autonomous Vehicle," in *ASME 2015 Dynamic System and Control Conference*, Columbus, Ohio, USA, 2015.

[90] B. Houska, H. J. Ferreau and M. Diehl, "ACADO toolkit—An open-source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods,* vol. 32, no. 3, pp. 298-312, 2011.

[91] B. Lau, K. O. Arras and W. Burgard, "Multi-model hypothesis group tracking and group size estimation," *International Journal of Social Robotics,* vol. 2, no. 1, pp. 19-30, 2010.

[92] D. Reid, "An algorithm for tracking multiple targets," *IEEE transactions on Automatic Control,* vol. 24, no. 6, pp. 843-854, 1976.

[93] Q. Wang and B. Ayalew, "Obstacle Filtering Algorithm for Control of an Autonomous Road Vehicle in Public Highway Traffic," in *Proceedings of the ASME 2016 Dynamic System and Control Conference (DSCC),* Minneapolis, MN., 2016.

[94] J. Nilsson, P. Falcone, M. Ali and J. Sjöberg, "Receding horizon maneuver generation for automated highway driving," *Control Engineering Practice,* vol. 41, pp. 124-133, 2015.

[95] U. Rosolia, F. Braghin, A. Alleyne and E. Sabbioni, "NLMPC for Real Time Path Following and Collision Avoidance," *SAE International Journal of Passenger Cars-Electronic and Electrical Systems,* vol. 8, no. 2, pp. 401-405, 2015.

[96] M. S. Menon and A. Ghosal, "Obstacle avoidance for hyper-redundant snake robots and one dimensional flexible bodies using optimization," in *14th World Congress in Mechanism and Machine Science*, Taipei, Taiwan, 2015.

[97] D. D. Salvucci, "Modeling driver behavior in a cognitive architecture," *The Journal of the Human Factors and Ergonomics Society,* vol. 48, no. 2, pp. 362-380, 2006.

[98] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems,* vol. 2, no. 4, pp. 303-314, 1989.

[99] L. Mihaylova, A. Y. Carmi, F. Septier, A. Gning, S. K. Pang and S. Godsill, "Overview of Bayesian sequential Monte Carlo methods for group and extended object tracking," *Digital Signal Processin,* vol. 25, pp. 1-16, 2014.

[100] K. Granstrom and M. Baum, "Extended object tracking: introduction, overview and applications," 2016. [Online]. Available: https://arxiv.org/abs/1604.00970.

[101] A. Gning, L. Mihaylova, S. Maskell, S. K. Pang and S. Godsill, "Group object structure and state estimation with evolving networks and Monte Carlo methods," *IEEE Transactions on Signal Processing,* vol. 59, no. 4, pp. 1383-1396, 2011.

[102] S. K. Pang, J. Li and S. J. Godsill, "Detection and tracking of coordinated groups," *IEEE Transactions on Aerospace and Electronic Systems* , vol. 47, no. 1, pp. 472-502, 2011.

[103] N. Petrov, L. Mihaylova, A. Gning and D. Angelova, "A novel sequential Monte Carlo approach for extended object tracking based on border parameterisation," in *In Information Fusion (FUSION), 2011 IEEE Proceedings of the 14th International Conference on*, 2011.

[104] B. Ristic and D. Salmond, "A study of a nonlinear filtering problem for tracking an extended target," in *2004 IEEE Seventh International Conference on Information Fusion*, 2004.

[105] K. Granström, C. Lundquist and U. Orguner, "Tracking rectangular and elliptical extended targets using laser measurements," in *Information Fusion (FUSION), 2011 IEEE Proceedings of the 14th International Conference on*, 2011.

[106] K. Granström, S. Reuter, D. Meissner and A. Scheel, "A multiple model PHD approach to tracking of cars under an assumed rectangular shape," in *Information Fusion (FUSION), 2014 17th IEEE International Conference on* , 2014.

[107] M. Baum and U. D. Hanebeck., "Shape tracking of extended objects and group targets with star-convex RHMs," in *Information Fusion (FUSION), 2011 IEEE Proceedings of the 14th International Conference on*, 2011.

[108] J. Lan and X. R. Li, "Tracking of extended object or target group using random matrix—part II: irregular object," in *In Information Fusion (FUSION), 2012 IEEE 15th International Conference on*, 2012.

[109] P. Willett, Y. Ruan and R. Streit, "PMHT: problems and some solutions," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 38, no. 3, pp. 738-754, 2002.

[110] B. Vo and W. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Transactions on signal processing,* vol. 54, no. 11, pp. 4091-4104, 2006.

[111] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica,* vol. 11, no. 5, p. 451–460, 1975.

[112] B.-N. Vo, S. Singh and A. Doucet, "Sequential Monte Carlo methods for multitarget filtering with random finite sets," *IEEE Transactions on Aerospace and electronic systems,* vol. 41, no. 4, pp. 1224-1245, 2005.

[113] S. Wender and K. Dietmayer, "3D vehicle detection using a laser scanner and a video camera," *IET Intelligent Transport Systems,* vol. 2, no. 2, pp. 105-112, 2008.

[114] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Autonomous Robots,* vol. 26, no. 2-3, pp. 123-139, 2009.

[115] T. Chen, T. B. Schon, H. Ohlsson and L. Ljung, "Decentralized particle filter with arbitrary state decomposition," *IEEE Transactions on Signal Processing,* vol. 59, no. 2, pp. 465-478, 2011.

[116] Q. Wang and B. Ayalew, "A multiple vehicle group modelling and computation framework for guidance of an autonomous road vehicle," in *Proccedings of the 2017 American Control Conference*, 2017.

[117] M. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on signal processing,* vol. 50, no. 2, pp. 174-188, 2002.

[118] N. E. Du Toit and J. W. Burdick, "Probabilistic collision checking with chance constraints," *IEEE Transactions on Robotics,* vol. 27, no. 4, pp. 809-815, 2011.

[119] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," 2008. [Online]. Available: http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3274/pdf/imm3274.pdf.

[120] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 1996.

[121] H.-P. Kriegel and M. Pfeifle, "Density-based clustering of uncertain data," in *In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005.

[122] A. F. Karr, Probability, New York: Springer-Verlag, 1993.

[123] "NGSIM Program Database," FHWA, 2016. [Online]. Available: https://www.its-rde.net/index.php/rdedataenvironment/10023.

[124] Q. Wang, B. Ayalew and T. Weiskircher, "Predictive Maneuver Planning for an Autonomous Vehicle in Uncertain Public Highway Traffic," *IEEE Transaction on Intelligent Transportation Systems,* Vols. (Submitted, in review), 2017.

[125] S. A. Tarim, S. Manandhar and T. Walsh, "Stochastic constraint programming: A scenario-based approach," *Constraints,* vol. 11, no. 1, pp. 53-80, 2006.

[126] L. Blackmore, H. Li and B. Williams, " A probabilistic approach to optimal robust path planning with obstacles," in *American Control Conference*, 2006.

[127] E. Balas, "Disjunctive programming," *Annals of Discrete Math,* vol. 5, pp. 3-51, 1979.

[128] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Systems,* vol. 36, no. 6, pp. 30-44, 2016.

[129] Q. Wang and B. Ayalew, "Constraint Tightening for the Probabilistic Collision Avoidance of Multi-Vehicle Groups in Uncertain Traffic," in *1st IEEE Conference on Control Technology and Applications*, Kohala Coast, Hawai'i, 2017.

[130] Q. Wang and B. Ayalew, "A Probabilistic Framework for Tracking the Formation and Evolution of Multi-Vehicle Groups in Uncertain Traffic," *IEEE Transactions on Intelligent Transportation Systems (Submitted, in review),* 2017.

[131] L. Blackmore, M. Ono, A. Bektassov and B. C. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE Transactions on Robotics,* vol. 26, no. 3, pp. 502-517, 2010.

[132] A. Mesbah, S. Streif, R. Findeisen and R. D. Braatz, "Stochastic nonlinear model predictive control with probabilistic constraints," in *American Control Conference*, 2014.

[133] C. Constable and D. C. Agnew, "Chapter 4: Multivariate Distributions. in Geophysical Data Analysis: Statistics," 2008. [Online]. Available: http://igppweb.ucsd.edu/~agnew/Courses/Sio223a/sio223a.chap4.pdf. [Accessed 9 3 2017].